AD-A078 392 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66158 SBIE-AD-E100 314 NL

1 OF 4
ADA 078392

66158

Book 2

November 1979

LEVEL II

SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM)

for

THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

Burroughs Corporation
Federal and Special Systems Group
Paoli, Pa. 19301

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 66158	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software Maintenance Manual for the Modular System Control Development Model (MSCDM) Book 2		5. TYPE OF REPORT & PERIOD COVERED FINAL Sep 76 - Nov 79 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DCA100-76-C-0083
9. PERFORMING ORGANIZATION NAME AND ADDRESS Burroughs Corporation Federal and Special Systems Group Paoli, PA 19301		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 33126 T&CCP 3012 Task 15203
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency Defense Communications Engineering Center 1860 Wiehle, Ave., Reston, VA 22090		12. REPORT DATE November 1979 13. NUMBER OF PAGES 284
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Program Listings	Modular Architecture
Loop Network	DCS System Control
Ring Network	Distributed Computer System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Program descriptions and listings of the Modular System Control Development Model utility programs and diagnostics are presented. This includes the Bootstrap Load Facility for the Ring Architecture.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(18) SBIE

(17) 66158

Book 2

(11) November 1979

(19) AD-E100 314

(12) 286

(6)

SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM). Book 2.

(15) DCA100-76-C-0083

for

THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

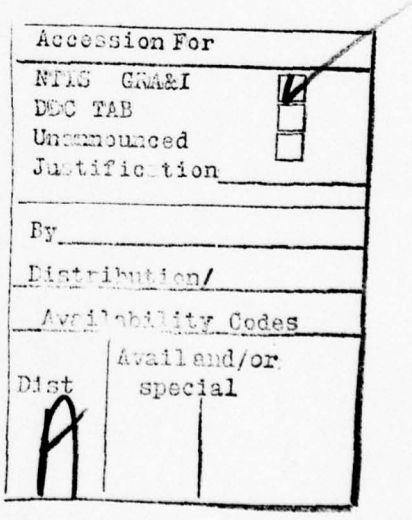
(9) Final rept. Sep 76-Nov 79.

Burroughs Corporation
Federal and Special Systems Group
Paoli, Pa. 19301

070 040

Foreword

This publication is the Software Maintenance Manual for the Modular System Control Development Model (MSCDM). This manual was prepared by the Burroughs Corporation and is submitted in accordance with the requirements of Contract DCA 100-76-C-0083.



MSCDM SOFTWARE MAINTENANCE MANUAL

MSCDM SOFTWARE MAINTENANCE MANUAL

Book 2

INTRODUCTION

The Modular System Control Development Model (MSCDM) consists of nine functional nodes: Station-to-Station Communications Interface (SSCI), Voice Service Quality Control (VSQC), Digital Service Quality Control (DSQC), Data Base Management Service (DBMS), Operator Control and Report Interface (OCRI), Baseband Signal Analysis and Wide Band Signal Analysis (BWBSA), Fault Isolation and Control Coordination (FIAC), Switch Data Collection and Analysis (SDCA), and Simulated Input Generator (SIG). Each of these nodes is implemented using microcomputer hardware and software, and node intercommunication is performed via a Burroughs loop architecture under control of a distributed master control program (DMCP).

Each node can communicate with any other node on the loop; however, the nodal software for the MSCDM application defines the flow of information in the system. For example, the OCRI terminal normally communicates with the DBMS node, which runs the User Language. The other ESM terminals communicate with the User Language via the loop 4-5 gateway Node 21 (SSCI).

A simulated input generator (SIG) generates inputs to the VSQC, DSQC and BWBSA, which communicate faults to the FIAC module. FIAC generates event reports to the OCRI and DBMS. The PDP 11/40 in loop 2 generates inputs to the SDCA which generates fault reports to the OCRI and DBMS. The DBMS, OCRI and FIAC communicate with the other loops via SSCI.

An LA36 DECWRITER is be used as the OCRI hard-copy terminal attached to node 25. A VT52 DECScope is be used as a local CRT terminal connected to the Program Development Unit (PDU).

The Digital Equipment Corporation System's software for the PDP-11/V03 is contained on 21 - Floppy diskettes conforming to DEC's RX01 Floppy drive format.

The MSCDM applications software which runs on the PDP-11/V03 and the LSI-11/2 nodes is contained on 9 - floppy diskettes.

References for the FORTRAN and MACRO-11 languages and the PDP-11 /V03 system used include documentation:

1. "RT-11 System Generation Manual"
2. "Introduction to RT11"
3. "RT-11 System User's Guide"
4. "RT-11 System Message Manual"
5. "PDP-11 MACRO Language Manual"
6. "PDP-11 FORTRAN Language Manual"
7. "Advanced Programmer's Guide"

2.0 UTILITY PROGRAM

2.1 PAGE PROGRAM DESCRIPTION

PGLOOP is a program that allows a user of MSCMD to get a hard copy print out of a source file stored on the PDP11/03's diskettes. Using the NODE 25's LA-36 printer this program is made up of three source files: PGLOOP, WRTLP AND PGMAC. This allows the LA-36 to act as a line printer by providing for FF, tab, and sequence numbers.

2.1.1 Using Page

1. Press Clear Switch on operator's panel
2. Insert Diskette #1 into Drive 0
3. Insert USER's diskette into Drive 1
4. Type .R PGLOOP
5. The CRT screen will then be formatted as follows

```
FILE [              ]
SEQN [      ]
```

6. Cursor is placed just right bracket of file line
7. Type your file to cpied, e.g., "123456.123"
8. Press RETURN
9. Type "Y or N" for sequence numbers
10. Place LA36 printer head just below fan fold of paper
11. Press RETURN

12. File will then begin to print, when complete the utility will return to step 5, enabling a new file to be typed in.
13. To stop PGLOOP type "END" when it asks for a file name.

NOTE: on files with types of ".LST, .MAP" PGLOOP will not ask for sequence numbers

NOTE: after the first file is printed, step 11 need not be done

2.1.2 Program Descriptions

2.1.2.1 Program PGLOOP (FORTRAN)

This program is the driver for the PGLOOP program; it opens the selected files, formats screen and pages the file for all the control characters.

2.1.2.1 Subroutine HWRITE (FORTRAN)

This subroutine is passed a byte that is written to the VT52.

2.1.2.3 Subroutine HEADER (FORTRAN)

This subroutine builds the line of information at the top of each page (i.e., file name, date, time).

2.1.2.3 Subroutine SPACE (FORTRAN)

This subroutine is passed a count of how many spaces to write.

2.1.2.4 Subroutine LINE (FORTRAN)

This subroutine is passed a count of how many line feeds to write.

2.1.2.5 Subroutine SEQNUM (FORTRAN)

This subroutine appends a sequence number onto a line when SEQN OPTION is selected.

2.1.2.6 Subroutine ACKNAK (FORTRAN)

This subroutine checks the response coming from node 25 for ACKs or NAKs; on a NAK the packet is retransmitted.

2.1.2.7 Subroutine WRITE (FORTRAN)

This subroutine forms the data into packets and writes them to the loop.

2.1.2.8 Subroutine MTAB (MACRO)

This subroutine calculates where the next tab character should be.

2.1.2.9 Subroutine INIT (MACRO)

This subroutine initializes the LIU at node 24.

2.1.2.10 Subroutine LIO (MACRO)

This is the interrupt handler for PGLOOP. It reads packets coming from node 25.

2.1.2.11 Subroutine RAM (MACRO)

This subroutine writes the read address for the node, into the ACRAM on the LIU.

2.1.2.12 Subroutine DABLE (MACRO)

This subroutine disables interrupts from the LIU.

2.1.2.13 Subroutine WTLOOP (MACRO)

This subroutine writes packets to the loop.

```
0001          PROGRAM PGLOOP
0002          BYTE CHAR,BUFFER(512),NAME(16),INBF,OUTBF
0003          INTEGER*4 NPAGE,NSEQ
0004          COMMON/BLK1/NAME
0005          COMMON/LOOP1/OUTBF(256),INBF(256)
0006          COMMON/LOOP2/IGOT,IACK,INAK,IFLT
0007          COMMON/SAVE/IPNT,IPCNT
0008          CALL DABLE
0009          CALL INIT(I1,I2,I3,I4,I5)
0010          IF(I1 .EQ. 1) GOTO 60
0012          STOP'ERROR--> BAD INITIALIZE'
0013       60 CALL RAM("005,4)
        C
0014      100 OUTBF(1)=0
0015          OUTBF(2)=6
0016          OUTBF(3)=3
0017          OUTBF(4)=7
0018          CALL WTLOOP(4)
0019          NLINE=0
0020          IPCNT=0
0021          IPNT=3
0022          NSEQ=1
0023          ISW=0
0024          JSW=0
0025          NPAGE=1
0026          NCHAR=0
0027          TYPE 4,("033,"007)
0028          TYPE 4,("033,"110)
0029          TYPE 4,("033,"112)
0030       4  FORMAT(1X,1A1,1A1)
0031          WRITE(7,11)
0032       11 FORMAT(//,1X,'FILE[               ]',/,1X,'SEQN[ ]')
0033          DO 12 I=1,16
0034       12 NAME(I)=32
0035          DO 14 I=1,2
0036          CALL HWRITE("033)
0037       14 CALL HWRITE("111)
0038          DO 15 I=1,6
0039          CALL HWRITE("033)
0040       15 CALL HWRITE("103)
0041          READ(7,40) JCHAR,(NAME(I), I=1,JCHAR)
0042       40 FORMAT(Q,16A1)
0043          DO 43 I=1,16
0044          IF(NAME(I) .NE. '.') GOTO 43
0046          IF(NAME(I+1) .EQ. 'L' .OR. NAME(I+1) .EQ. 'M' .AND.
            1  NAME(I+3) .EQ. 'P') JSW=1
0048       43 CONTINUE
0049          IF(JSW .EQ. 1) GOTO 28
0051          IF(NAME(1) .EQ. 'E' .AND. NAME(2) .EQ. 'N' .AND. NAME(3) .EQ. 'I')
            2  GOTO 6
0053          DO 41 I=1,5
0054          CALL HWRITE("033)
0055       41 CALL HWRITE("103)
0056          READ(7,42) IT
```

```
0057    42  FORMAT(1A1)
0058        IF(IT .EQ. 'Y') ISW=1
      C
0060    28  CALL ASSIGN(6,NAME)
0061        DEFINE FILE 6(300,256,U,IREC)
0062        IREC=1
0063        CALL WRITE("007)
      C
0064    1   READ(6'IREC,END=5,ERR=2) (BUFFER(I), I=1,512)
0065        FIND(6'IREC)
0066        IF(IREC .GT. 2 .OR. JSW .EQ. 1) GOTO 9
0068           CALL HEADER(0,NPAGE)
0069        NPAGE=2
0070    9   DO 7 I=1,512
0071        IF(BUFFER(I) .EQ. "000 .AND. BUFFER(I+1) .EQ. "000) GOTO 1
0073        IF(BUFFER(I) .NE. "014) GOTO 17
0075        NPAGE=NPAGE+1
0076        IF(JSW .NE. 1) GOTO 30
0078        IL=66-NLINE
0079        CALL LINE(IL)
0080        NLINE=0
0081        GOTO 17
0082    30  IL=57-NLINE
0083        CALL LINE(IL)
0084         IF(BUFFER(I) .EQ. "000 .AND. BUFFER(I+1) .EQ. "000) GOTO 1
0086         GOTO 18
0087    17  IF(BUFFER(I-1) .EQ. "012 .AND. ISW .EQ. 2) CALL SEQNUM(NSEQ)
0089        IF(ISW .EQ. 1 .AND. NSEQ .EQ. 1) CALL SEQNUM(1)
0091        IF(ISW .EQ. 1) ISW=2
0093        IF(BUFFER(I) .NE. "011) GOTO 8
0095         CALL MTAB(NCHAR,NSPACE)
0096         CALL SPACE(NSPACE-NCHAR)
0097         NCHAR=0
0098         GOTO 7
0099    8   CALL WRITE(BUFFER(I))
0100        NCHAR=NCHAR+1
0101    29  IF(BUFFER(I) .NE. "012) GOTO 7
0103        NSEQ=NSEQ+1
0104        NLINE=NLINE+1
0105        NCHAR=0
0106        IF(NLINE .LT. 57 .OR. JSW .EQ. 1) GOTO 7
0108    18  CALL HEADER(1,NPAGE)
0109        NPAGE=NPAGE+1
0110        NLINE=0
0111    7   CONTINUE
0112        GOTO 1
      C
0113    5   IL=61-NLINE
0114        CALL LINE(IL)
0115        IF(JSW .EQ. 1) CALL LINE(5)
0117        CALL CLOSE(6)
0118        IF(IPNT .EQ. 3) GOTO 61
0120        DO 62 J=IPNT,130
0121        CALL WRITE(0)
```

```
0122     62  CONTINUE
0123     61  GOTO 100
0124     6   CALL DABLE
0125         STOP 'PAGE'
0126     2   WRITE(7,3)
0127     3   FORMAT(/,1X,'ERROR IN FILE NAME-RETYPE')
0128         GOTO 100
0129         END
      C******
```

```
FORTRAN IV     V02.1-1    Tue 27-Mar-79 20:24:18                PAGE 001

0001          SUBROUTINE HWRITE(NBYTE)
0002          CALL IPOKEB("177566,NBYTE)
0003        1 IF(IPEEK("177564) .NE. "200) GOTO 1
0005          RETURN
0006          END
      C*******
```

```
0001          SUBROUTINE HEADER(ITYPE,IPAGE)
0002          BYTE JDATE(9),JTIME(8),TITLE(22),JPAGE(4)
0003          INTEGER*4 IPAGE
0004          REAL*8 TI(3),P1
0005          EQUIVALENCE (TITLE,TI),(PAGE,P1)
0006          COMMON/BLK1/NAME
0007          BYTE NAME(16),PAGE(5)
0008          DATA TI(1),TI(2),TI(3)/'DECWRITE','R     PR','OGRAM:  '/
0009          DATA P1/'PAGE    '/
0010          CALL DATE(JDATE)
0011          CALL TIME(JTIME)
      C
0012          IF(ITYPE .EQ. 0) CALL LINE(2)
0014          IF(ITYPE .EQ. 1) CALL LINE(6)
      C
0016          CALL SPACE(5)
0017          DO 40 I=1,22
0018          IF(TITLE(I) .LE. 90 .AND. TITLE(I) .GE. 65) GOTO 44
0020          CALL WRITE(TITLE(I))
0021          GOTO 40
0022       44 CALL WRITE(TITLE(I)+32)
0023       40 CONTINUE
0024          CALL SPACE(4)
0025          DO 45 I=1,16
0026          CALL WRITE(NAME(I))
0027       45 CONTINUE
0028          CALL SPACE(15)
0029          DO 50 I=1,9
0030           CALL WRITE(JDATE(I))
0031       50 CONTINUE
0032          CALL SPACE(10)
0033          DO 60 I=1,8
0034          CALL WRITE(JTIME(I))
0035       60 CONTINUE
0036          CALL SPACE(10)
0037          DO 80 I=1,5
0038           CALL WRITE(PAGE(I))
0039       80 CONTINUE
0040          ENCODE(4,22,JPAGE) IPAGE
0041       22 FORMAT(4I)
0042          DO 23 I=2,4
0043          IF(JPAGE(I) .LE. 57 .AND. JPAGE(I) .GE. 48) GOTO 25
0045          JPAGE(I)=48
0046       25 CALL WRITE(JPAGE(I))
0047       23 CONTINUE
0048          CALL LINE(3)
      C
0049          RETURN
0050          END
      C*******
```

```
FORTRAN IV     V02.1-1    Tue 27-Mar-79 20:24:46              PAGE 001

0001          SUBROUTINE SPACE(ICOUNT)
0002          DO 20 I=1,ICOUNT
0003            CALL WRITE(32)
0004       20   CONTINUE
0005          RETURN
0006          END
      C*******
```

```
FORTRAN IV     V02.1-1    Tue 27-Mar-79 20:24:58          PAGE 001

0001          SUBROUTINE LINE(JCOUNT)
0002          CALL WRITE(13)
0003          DO 30 I=1,JCOUNT
0004           CALL WRITE(10)
0005       30  CONTINUE
0006          RETURN
0007          END
      C******
```

```
0001          SUBROUTINE SEQNUM(INUM)
0002          INTEGER*4 INUM
0003          BYTE JNUM(4)
0004          ENCODE(4,2,JNUM) INUM
0005      2   FORMAT(4I)
0006          DO 3 I=1,4
0007          IF(JNUM(I) .LE. 57 .AND. JNUM(I) .GE. 48) GOTO 4
0009          JNUM(I)=32
0010      4   CALL WRITE(JNUM(I))
0011      3   CONTINUE
0012          CALL SPACE(4)
0013          RETURN
0014          END
```

```
FORTRAN IV      V02.1-11   Mon 01-Oct-79 12:00:39              PAGE 001

0001          SUBROUTINE ACKNAK(IT)
0002          COMMON/LOOP2/IGOT,IACK,INAK,IFLT
      C
0003      100 DO 1,I=1,40
0004          DO 4 J=1,1000
0005          IF(IGOT .EQ. 1) GOTO 2
0007      4   CONTINUE
0008      1   CONTINUE
0009      2   IF(IACK .NE. 1) GOTO 3
0011          IT=1
0012          GOTO 999
0013      3   IT=-1
0014      999 IGOT=0
0015          IACK=0
0016          INAK=0
0017          IFLT=0
0018          RETURN
0019          END
```

```
0001          SUBROUTINE WRITE(IBYTE)
0002          BYTE INBF,OUTBF,IBYTE
0003          COMMON/LOOP1/OUTBF(256),INBF(256)
0004          COMMON/SAVE/IPNT,IPCNT
      C
0005          OUTBF(IPNT)=IBYTE
0006          IPNT=IPNT+1
0007          IF(IPNT .NE. 131) GOTO 999
0009          IPCNT=IPCNT+1
0010          IF(IPCNT .EQ. 100) IPCNT=1
0012          OUTBF(1)=IPCNT
0013          OUTBF(2)=7
0014          OUTBF(131)=003
0015          OUTBF(132)=007
0016          IECNT=1
0017     1    CALL WTLOOP(132)
0018          CALL ACKNAK(IST)
0019          IF(IST .EQ. 1 ) GOTO 900
0021          IECNT=IECNT+1
0022          IF(IECNT .LT. 3) GOTO 1
0024          CALL DABLE
0025          STOP'ERROR--> NO RESPONSE FROM NODE'
0026    900   IPNT=3
0027    999   RETURN
0028          END
      C*******
```

```
                                        .TITLE PRINTER MACRO'S
                                        .IDENT  /V1.0/
                                        .GLOBL  WTLOOP,INIT,RAM,DABLE,MTAB
000000                                  .PSECT

        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        000002                          ARG1=   2
        000004                          ARG2=   4
        000006                          ARG3=   6
        000010                          ARG4=   10
        000012                          ARG5=   12
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1


                                        ;TAB CALCULATOR

000000  012500                  MTAB:   MOV     (R5)+,  R0
000002  005001                          CLR     R1
000004  113501                          MOVB    @(R5)+, R1
000006  105201                  ADD:    INCB    R1
000010  132701  000007                  BITB    #7,R1
000014  001374                          BNE     ADD
000016  010135                          MOV     R1,@(R5)+
000020  000207                          RTS     PC
```

```
                                        ;INITIALIZE LIU
                                        ;
000022  012737  010420  172416  INIT:   MOV     #4368., @#OPREG         ;MOD S AT
000030  012737  004414  172416          MOV     #4414,  @#OPREG         ;SET SWITCHES
000036  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000042  100375                          BPL     .-4                     ;LOOP UNTIL
000044  105037  172414                  CLRB    @#CSR
000050  012737  004540  172416          MOV     #4540,  @#OPREG         ;RESET SWITCHS
000056  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000062  100375                          BPL     .-4                     ;LOOP UNTIL
000064  105037  172414                  CLRB    @#CSR
000070  012700  000124                  MOV     #124, R0                ;INTERRUPT VECTOR
000074  012720  000764'                 MOV     #LIO, (R0)+             ;ADDRESS OF LIO
000100  012710  000340                  MOV     #340, (R0)              ;PRI =340
000104  005003                          CLR     R3                      ;CLEAR ERROR COUNT
000106  005067  001622                  CLR     ERAM
000112  005067  001620                  CLR     EBUF
000116  005067  001616                  CLR     ESTAT
                                        ;
                                        ;INITIALIZE ACRAM
                                        ;
000122  005004                  LINIT:  CLR     R4                      ;CLEAR LOOP COUNT
000124  052737  000001  172414          BIS     #BIT00, @#CSR           ;INIT LIU
000132  000240                          NOP
000134  000240                          NOP
000136  042737  000001  172414          BIC     #BIT00, @#CSR           ;CLEAR LIU
000144  012737  010402  172416          MOV     #4354., @#OPREG         ;LDADR
000152  012737  004400  172416          MOV     #2304., @#OPREG         ;ADDRESS=0
000160  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000164  100375                          BPL     .-4                     ;NO RETRY
000166  105037  172414                  CLRB    @#CSR
000172  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
000200  012700  000400                  MOV     #256., R0               ;COUNTER
000204  012737  004407  172416  1$:     MOV     #2311., @#OPREG         ;WRITE A NULL
000212  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000216  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
000220  105037  172414                  CLRB    @#CSR
000224  077011                          SOB     R0,     1$              ;LOOP UNTIL ZERO
000226  012700  000400                  MOV     #256.,  R0              ;COUNTER
000232  012737  001400  172416  2$:     MOV     #768.,  @#OPREG         ;RD
000240  105737  172414                  TSTB    @#CSR                   ;GOOD RD
000244  100375                          BPL     .-4                     ;NO RETRY
000246  105037  172414                  CLRB    @#CSR
000252  013701  172416                  MOV     @#IOBUF, R1             ;FETCH DATA
000256  042701  177760                  BIC     #177760, R1             ;CLEAR BITS
000262  022701  000007                  CMP     #7,      R1             ;DATA=NULL
000266  001403                          BEQ     3$                      ;YES
000270  005204                          INC     R4                      ;NO REPORT IT
000272  005267  001436                  INC     ERAM                    ;RAM ERROR
000276  077023                  3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO
                                        ;
                                        ;INITIALIZE BUFFERS
                                        ;
000300  012701  010410          INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
000304  012700  010440                  MOV     #10440, R0              ;SEL INBUF0
```

```
 58 000310  004767  000106                    JSR     PC,     ZEROBP          ;SET POINTER=0
 59 000314  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
 60 000322  105737  172414                    TSTB    @#CSR                   ;GOOD READ
 61 000326  100375                            BPL     .-4                     ;NO RETRY
 62 000330  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 63 000334  012701  010610          INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
 64 000340  012700  010640                    MOV     #10640, R0              ;SEL INBUF1 COMMAND
 65 000344  004767  000052                    JSR     PC,     ZEROBP          ;POINTER=0
 66 000350  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
 67 000356  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 68 000362  100375                            BPL     .-4                     ;NO RETRY
 69 000364  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 70 000370  012701  010510          OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
 71 000374  012700  010540                    MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
 72 000400  004767  000016                    JSR     PC,     ZEROBP
 73 000404  012701  010710          OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
 74 000410  012700  010740                    MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
 75 000414  004767  000002                    JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
 76 000420  000464                            BR      STAT                    ;GO CLEAR STATUS
 77 000422  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
 78 000426  012737  001400  172416            MOV     #1400,  @#OPREG         ;RD
 79 000434  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 80 000440  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
 81 000442  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 82 000446  013702  172416                    MOV     @#IOBUF, R2             ;FETCH POINTER
 83 000452  042702  177400                    BIC     #177400, R2             ;CLEAR MST BYTE
 84 000456  010037  172416                    MOV     R0,     @#OPREG         ;SEL BUFFER
 85 000462  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
 86 000466  001412                            BEQ     5$
 87 000470  012737  001400  172416            MOV     #1400,  @#OPREG         ;NO FALSE RD
 88 000476  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 89 000502  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
 90 000504  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 91 000510  005302                            DEC     R2                      ;POINTER-1
 92 000512  000763                            BR      4$
 93 000514  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
 94 000520  012737  001400  172416            MOV     #1400,  @#OPREG         ;RD
 95 000526  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 96 000532  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
 97 000534  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 98 000540  013702  172416                    MOV     @#IOBUF, R2             ;FETCH NEW POINTER
 99 000544  042702  177400                    BIC     #177400, R2             ;CLEAR MST BYTE
100 000550  022702  000000                    CMP     #0,     R2              ;POINTER=0
101 000554  001403                            BEQ     6$
102 000556  005204                            INC     R4                      ;REPORT ERROR
103 000560  005267  001152                    INC     EBUF                    ;BUFFER ERROR
104 000564  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
105 000570  000207                            RTS     PC                      ;RETURN
106
107                                         ;INITIALIZE STATUS
108
109 000572  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR:R
110 000600  012737  002400  172416            MOV     #1280., @#OPREG         ;RS(FALSE)
111 000606  012737  002400  172416            MOV     #1280., @#OPREG         ;RS
112 000614  013700  172416                    MOV     @#OPREG, R0             ;FETCH STATUS
113 000620  042700  177400                    BIC     #177400, R0             ;CLEAR MST BYTE
114 000624  022700  000000                    CMP     #0,     R0              ;STATUS=0 ?
```

```
115 000630  001403                        BEQ     7$
116 000632  005204                        INC     R4                      ;STATUS BAD REPORT
117 000634  005267  001100                INC     ESTAT                   ;STATUS ERROR
118 000640  022704  000000          7$:   CMP     #0,     R4              ;WERE THERE ERRORS
119 000644  001406                        BEQ     8$                      ;NO GOOD INIT
120 000646  005203                        INC     R3                      ;ERROR +1
121 000650  022703  000005                CMP     #5,     R3              ;ERROR LIMIT YET
122 000654  001421                        BEQ     9$                      ;YES RETURN A 1(ERROR)
123 000656  000167  177240                JMP     LINIT                   ;NO RESTART VINIT
124 000662  012775  000001  000002  8$:   MOV     #1,     @ARG1(R5)       ;GOOD INIT
125 000670  005075  000004                CLR     @ARG2(R5)
126 000674  005075  000006                CLR     @ARG3(R5)
127 000700  005075  000010                CLR     @ARG4(R5)
128 000704  005075  000012                CLR     @ARG5(R5)
129 000710  052737  040000  172414        BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
130 000716  000207                        RTS     PC
131 000720  012775  177777  000002  9$:   MOV     #-1,    @ARG1(R5)       ;BAD INIT
132 000726  016775  001002  000004        MOV     ERAM,   @ARG2(R5)
133 000734  016775  000776  000006        MOV     EBUF,   @ARG3(R5)
134 000742  016775  000772  000010        MOV     ESTAT,  @ARG4(R5)
135 000750  010075  000012                MOV     R0,     @ARG5(R5)
136 000754  042737  040000  172414        BIC     #BIT14, @#CSR           ;DISABLE LIU
137 000762  000207                        RTS     PC
138                                       .ENABLE LSB
```

```
 1
 2                                          ;LIU HANDLER
 3
 4 000764  010046                  LIO:    MOV     R0,     -(SP)           ;SAVE REGISTERS
 5 000766  010146                          MOV     R1,     -(SP)
 6 000770  010246                          MOV     R2,     -(SP)
 7 000772  010346                          MOV     R3,     -(SP)
 8 000774  010446                          MOV     R4,     -(SP)
 9 000776  010546                          MOV     R5,     -(SP)
10 001000  005067  000726                  CLR     DATA                    ;CLEAR VAR.
11 001004  005737  172414                  TST     @#CSR                   ;INT
12 001010  100077                          BPL     RTI$                    ;NO RETURN
13 001012  042737  040000  172414          BIC     #BIT14, @#CSR           ;DISABLE
14 001020  012737  010400  172416          MOV     #4352., @#OPREG         ;%CR : RS(0)
15 001026  012737  002400  172416          MOV     #1280., @#OPREG         ;RD
16 001034  013701  172416                  MOV     @#IOBUF, R1
17 001040  042701  177400                  BIC     #177400, R1             ;CLEAR UNUSED BITS
18
19                                          ;BUFFER OVERFLOW
20
21 001044  132701  000004          B2$:    BITB    #BIT02, R1              ;OV-FL
22 001050  001406                          BEQ     B3$
23 001052  012702  010410                  MOV     #4360., R2              ;COM=SEL INBUF 0
24 001056  012703  010440                  MOV     #4384., R3              ;COM=RDBUFPNT
25 001062  004767  000146                  JSR     PC, EMBF                ;EMPTY BUFFER
26 001066  132701  000010          B3$:    BITB    #BIT03, R1              ;OV-FL
27 001072  001406                          BEQ     B0$
28 001074  012702  010610                  MOV     #4488., R2              ;COM=SEL INBUF 1
29 001100  012703  010640                  MOV     #4512., R3              ;COM=RDBUFPNT
30 001104  004767  000124                  JSR     PC, EMBF
31
32                                          ;BUFFERS FULL
33
34 001110  132701  000001          B0$:    BITB    #BIT00, R1              ;BUFFER 0 FULL
35 001114  001415                          BEQ     B1$                     ;NO CK 1
36 001116  012702  010410                  MOV     #4360., R2              ;YES, COM=RDBUFADR
37 001122  012703  010440                  MOV     #4384., R3              ;COM=SEL INBUF0
38 001126  012704  000001                  MOV     #BIT00, R4              ;WHICH CRC BIT
39 001132  004767  000076                  JSR     PC, EMBF                ;GO EMPTY AND LOAD
40 001136  005767  000570                  TST     DATA
41 001142  100402                          BMI     B1$
42 001144  004767  000264                  JSR     PC, PARSE               ;GO CHECK PACKET
43 001150  132701  000002          B1$:    BITB    #BIT01, R1              ;BUFFER 1 FULL
44 001154  001415                          BEQ     RTI$                    ;NO LOOP BACK
45 001156  012702  010610                  MOV     #4488., R2              ;YES,COM=RDBUFADR
46 001162  012703  010640                  MOV     #4512., R3              ;COM=SEL INBUF1
47 001166  012704  000002                  MOV     #BIT01, R4              ;WHICH CRC BIT
48 001172  004767  000036                  JSR     PC,     EMBF            ;EMPTY AND LOAD
49 001176  005767  000530                  TST     DATA
50 001202  100402                          BMI     RTI$
51 001204  004767  000224                  JSR     PC, PARSE               ;GO CHECK PACKET
52
53 001210  012605                  RTI$:   MOV     (SP)+, R5               ;RESTORE REGISTERS
54 001212  012604                          MOV     (SP)+, R4
55 001214  012603                          MOV     (SP)+, R3
56 001216  012602                          MOV     (SP)+, R2
57 001220  012601                          MOV     (SP)+, R1
```

```
    58 001222  012600                          MOV     (SP)+, R0
    59 001224  052737  040000  172414          BIS     #BIT14, @#CSR
    60 001232  000002                          RTI
    61
    62                                         ;EMPTY BUFFERS
    63
    64 001234  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG         ;RS (CRC)
    65 001242  012737  002400  172416          MOV     #1280., @#OPREG         ;RS CMD
    66 001250  013705  172416                  MOV     @#IOBUF, R5             ;FETCH STATUS
    67 001254  130405                          BITB    R4, R5                  ;CRC OK
    68 001256  001003                          BNE     5$                      ;YES CONTINUE
    69 001260  012767  177777  000444          MOV     #-1,    DATA            ;FLAG BAD CRC
    70 001266  012737  000400' 172410  5$:     MOV     #INBF, @#BAR            ;ADDRESS TO WRITE
    71 001274  010237  172416                  MOV     R2,     @#OPREG         ;RDBUFADR
    72 001300  012737  001400  172416          MOV     #768.,  @#OPREG         ;READ POINTER
    73 001306  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    74 001312  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    75 001314  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    76 001320  013704  172416                  MOV     @#IOBUF, R4             ;READ POINTER
    77 001324  042704  177400                  BIC     #177400, R4             ;CLEAR UNUSED BITS
    78 001330  005404                  6$:     NEG     R4                      ;2'S COMP
    79 001332  010437  172412                  MOV     R4,  @#WCR              ;BYTE COUNT
    80 001336  010337  172416                  MOV     R3,     @#OPREG         ;SEL BUFFER
    81 001342  012737  001400  172416          MOV     #768.,  @#OPREG         ;FALSE RD
    82 001350  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    83 001354  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    84 001356  105037  172414                  CLRB    @#CSR
    85 001362  012737  021000  172416          MOV     #8704., @#OPREG         ;FIRE DMA RD
    86 001370  000240                          NOP                             ;DELAY
    87 001372  105737  172414                  TSTB    @#CSR                   ;GOOD DMA
    88 001376  100403                          BMI     7$                      ;YES
    89 001400  012767  177777  000324          MOV     #-1,  DATA              ;BAD DMA XFER
    90 001406  105037  172414          7$:     CLRB    @#CSR                   ;CLEAR CSR
    91 001412  012737  004400  172416          MOV     #2304., @#OPREG         ;RESET POINTER TO 255
    92 001420  105737  172414                  TSTB    @#CSR                   ;GOOD WD
    93 001424  100375                          BPL     .-4                     ;NO RETRY
    94 001426  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    95 001432  000207                  8$:     RTS     PC
    96                                         .ENABLF LSB
    97
    98 001434  012702  000400'         PARSE:  MOV     #INBF, R2               ;ADDRESS OF PACKET
    99 001440  116203  000001                  MOVB    1(R2), R3               ;FETCH COMMAND
   100 001444  122703  000002          1$:     CMPB    #2,     R3              ;=ACK
   101 001450  001007                          BNE     2$
   102 001452  012767  000001  000000'         MOV     #1,     IGOT            ;YES REPORT
   103 001460  012767  000001  000002'         MOV     #1,     IACK
   104 001466  000207                          RTS     PC                      ;RETURN
   105 001470  122703  000003          2$:     CMPB    #3,     R3              ;=NAK
   106 001474  001007                          BNE     3$
   107 001476  012767  000001  000000'         MOV     #1,     IGOT            ;YES REPORT
   108 001504  012767  000001  000004'         MOV     #1,     INAK            ;FLAG
   109 001512  000207                          RTS     PC
   110 001514  012767  000001  000006' 3$:     MOV     #1,     IFLT
   111 001522  012767  000001  000000'         MOV     #1,     IGOT
   112 001530  000207                          RTS     PC
   113
```

```
                                        ;WRITE RAM
001532  017500  000002          RAM:    MOV     @ARG1(R5), R0                   ;ADDRESS
001536  017501  000004                  MOV     @ARG2(R5), R1                   ;WRITE DATA
001542  012737  010402  172416          MOV     #4354., @#OPREG                 ;SEL LDACTR
001550  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
001554  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
001560  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001564  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001566  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001572  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
001600  062701  004400                  ADI     #2304., R1                      ;WD/DATA (CMD)
001604  010137  172416                  MOV     R1,     @#OPREG                 ;WRITE
001610  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001614  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001616  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001622  000207                          RTS     PC

                                        ;DISABLE INTERRUPTS

001624  042737  040000  172414  DABLE:  BIC     #BIT14, @#CSR                   ;DIABLE LIU
001632  000207                  RTS     PC

                                        ;WRITE LOOP

001634  017502  000002          WTLOOP: MOV     @ARG1(R5), R2                   ;BYTE COUNT
001640  012737  000000' 172410          MOV     #OUTBF, @#BAR                   ;BUS ADDRESS
001646  005402                          NEG     R2                              ; 2'S COMP COUNT
001650  010237  172412                  MOV     R2,     @#WCR                   ;COUNT
001654  012737  010540  172416          MOV     #4448., @#OPREG                 ;OB0 COMMAND
001662  012737  024000  172416          MOV     #10240., @#OPREG                ;DMA GO
001670  000240                          NOP                                     ;INTERFACE TIME
001672  105737  172414                  TSTB    @#CSR                           ;DMA OK
001676  100401                          BMI     CMD                             ;YES
001700  000000                          HALT
001702  012737  010420  172416  CMD:    MOV     #4368., @#OPREG                 ;MODSTAT COMMAND
001710  012737  004421  172416          MOV     #4421,  @#OPREG                 ;BUFFERS FULL
001716  105737  172414                  TSTB    @#CSR                           ;GOOD WRITE
001722  100375                          BPL     .-4                             ;NO LOOP UNTIL
001724  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001730  000207                          RTS     PC


001732                                  .PSECT

001732  000000          DATA:   .WORD   0
001734                  ERAM:   .BLKW
001736                  EBUF:   .BLKW
001740                  ESTAT:  .BLKW

000000                          .PSECT  LOOP1,RW,D,GBL,REL,OVR
000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.

000000                          .PSECT  LOOP2,RW,D,GBL,REL,OVR
000000                  IGOT:   .BLKW
```

```
     58 000002                          IACK:   .BLKW
     59 000004                          INAK:   .BLKW
     60 000006                          IFLT:   .BLKW
     61
     62
     63          000001                         .END
```

```
PRINTER MACRO'S        MACRO V03.01 27-MAR-79 19:51:50 PAGE 4-2
SYMBOL TABLE

ADD     000006R         BIT04 = 000020          CMD     001702R         INAK    000004R     003 OTBUF0  000370R
ARG1  = 000002          BIT05 = 000040          CSR   = 172414          INBF    000400R     002 OTBUF1  000404R
ARG2  = 000004          BIT06 = 000100          DABLE   001624RG        INBUF0  000300R         OUTBF   000000R     002
ARG3  = 000006          BIT07 = 000200          DATA    001732R         INBUF1  000334R         PARSE   001434R
ARG4  = 000010          BIT08 = 000400          EBUF    001736R         INIT    000022RG        RAM     001532RG
ARG5  = 000012          BIT14 = 040000          EMBF    001234R         IOBUF = 172416          RTI$    001210R
BAR   = 172410          BIT15 = 100000          ERAM    001734R         LINIT   000122R         STAT    000572R
BIT00 = 000001          B0$     001110R         ESTAT   001740R         LIO     000764R         WCR   = 172412
BIT01 = 000002          B1$     001150R         IACK    000002R     003 MTAB    000000RG        WTLOOP  001634RG
BIT02 = 000004          B2$     001044R         IFLT    000006R     003 OPREG = 172416          ZEROBP  000422R
BIT03 = 000010          B3$     001066R         IGOT    000000R     003

. ABS.  000000      000
        001742      001
LOOP1   001000      002
LOOP2   000010      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PGMAC=DK:PGMAC
```

```
ASS DX0: DK:
R LINK
DX1:PGLOOP=DX1:PGLOOP/C
DX1:WRTIF/C
DX1:PGMAC//

ASS DX1: DK:
```

2.2 FORMAT PROGRAM

2.2.1 Using the MSGCON Program

Thie program is found on diskette #3A, and is run by typing "RUN MSGCON" after the PDU power is up as per the USER MANUAL. At the prompt "ENTER INPUT FILE*", type the desired input source file (must be contained on DX1:, diskette #3A). At prompt "ENTER OUTPUT FILE*", type the name of the file to be formed with 80 byte records. This formats the file and stops.

2.2.2 Program Description

The program reads an input file name and an output file name from the user and opens the input file as read only. The program reads blocks of the file and writes 80 byte records to the output file for each line of text of the input file.

```
      C     ******************************
      C     ******************************
      C     **                          **
      C     **   THIS PROGRAM WILL      **
      C     **   CONVERT A FILE CREATED **
      C     **   EDIT INTO A FILE WITH  **
      C     **   80 BYTE RECORDS        **
      C     **                          **
      C     ******************************
      C     ******************************
0001          PROGRAM MSGCON
0002          BYTE IBUF,TBUF,COUNT
0003          INTEGER STARTB,ENDB,TSTART,TEND,NREC,MREC,RECNUM
0004          DIMENSION IBUF(512),TBUF(80)
0005          WRITE(7,2)
0006        2 FORMAT('$','ENTER INPUT FILE ')
0007          CALL ASSIGN(8,,-1)
0008          WRITE(7,4)
0009        4 FORMAT('$','ENTER OUTPUT FILE ')
0010          CALL ASSIGN(9,,-1)
0011          DEFINE FILE 8(200,256,U,NREC)
0012          DEFINE FILE 9(500,40,U,MREC)
0013          NREC=1
0014          MREC=1
0015          STARTB=1
0016       10 READ (8'NREC,END=500)(IBUF(I),I=1,512)
0017          I=1
0018          COUNT=0
0019       15 CONTINUE
0020       20     IF (IBUF(I) .EQ. "15) GOTO 30
0022          IF (I .GE. 512) GOTO 40
0024          I=I+1
0025          GOTO 20
0026       30 ENDB=I-1
0027          WRITE (9'MREC,END=500)(IBUF(J),J=STARTB,ENDB)
0028          STARTB=ENDB+3
0029          I=I+1
0030          GOTO 15
0031       40 I=0
0032          DO 44 J=1,80
0033            TBUF(J)=0
0034       44 CONTINUE
0035          DO 50 J=STARTB,512
0036            I=I+1
0037       50 TBUF(I)=IBUF(J)
0038          TSTART=I+1
0039          READ(8'NREC,END=500)(IBUF(I),I=1,512)
0040          I=1
0041       60 IF (IBUF(I) .EQ. "15) GOTO 70
0043          I=I+1
0044          GOTO 60
0045       70 K=0
0046          TEND=TSTART+I-2
0047          DO 80 J=TSTART,TEND
```

```
0048            K=K+1
0049     80   TBUF(J)=IBUF(K)
0050          WRITE (9'MREC,END=500)(TBUF(J),J=1,TEND)
0051          STARTB=I+2
0052          I=I+1
0053          GOTO 15
0054     500  CONTINUE
0055          DO 550 J=1,80
0056     550  TBUF(J)=0
0057     560  WRITE(9'MREC,END=600)(TBUF(J),J=1,80)
0058          GOTO 560
0059     600  CALL CLOSE(8)
0060          CALL CLOSE(9)
0061          STOP'CONVERT'
0062          END
```

2.3 STATUS PROGRAM

2.3.1 Description and Use of Status Program (BDSTAT):

The program is run by first power-up the PDU as per Section 1 and typing "RUN BDSTAT", the program opens a file called "STATUS. MAS" and writes 1015 records contain green channel and link data, status records, later to by renamed to "STATUS. DAT" which is used by MSCDM user language. This program is only used when the STATUS. MAS is corupted or deleted.

```
AFORTRAN IV      V02.1-11   Mon 01-Oct-79 12:00:55              PAGE 001

0001          PROGRAM BDSTAT
      C        BUILD STATUS FILE FOR MSCDM
0002          BYTE BD4,OUT(20)
0003          INTEGER*4 ID4
0004          REAL*8 I1,I2,I3
0005          REAL*4 I4
0006          DATA I1,I2,I3/'AAAAAAAA','BBBBBBBB','  111110'/
0007          DATA I4/'XXXX'/
0008          COMMON/IM/BD4(4)
0009          CALL ASSIGN(1,'STATUS.MAS')
0010          DEFINE FILE 1(1014,10,U,IREC)
0011          IREC=1
      C
0012          DO 100 J=1,500
0013          JJ=J
0014          IS=JICVT(JJ,ID4)
0015          CALL IMAGE(ID4)
0016          ENCODE(8,101,OUT(1)) I1
0017      101 FORMAT(A8)
0018          DO 102 I=1,4
0019      102 OUT(I+4)=BD4(I)
0020          OUT(9)=' '
0021          DO 103 I=1,4
0022      103 OUT(I+9)=BD4(I)
0023          OUT(14)=' '
0024          OUT(15)='3'
0025          DO 104 I=16,20
0026      104 OUT(I)=' '
0027          WRITE(1'J)(OUT(I),I=1,20)
      D       WRITE(7,105)(OUT(I),I=1,20)
0028      105 FORMAT(1X,20A1)
0029      100 CONTINUE
      C
0030          DO 200 J=501,1000
0031          JJ=J
0032          IS=JICVT(JJ,ID4)
0033          CALL IMAGE(ID4)
0034          ENCODE(8,201,OUT(1)) I2
0035      201 FORMAT(A8)
0036          DO 202 I=1,4
0037      202 OUT(I+4)=BD4(I)
0038          OUT(9)=' '
0039          DO 203 I=1,4
0040      203 OUT(I+9)=BD4(I)
0041          OUT(14)=' '
0042          OUT(15)='3'
0043          DO 204 I=16,20
0044      204 OUT(I)=' '
0045          WRITE(1'J)(OUT(I),I=1,20)
      D       WRITE(7,205)(OUT(I),I=1,20)
0046      205 FORMAT(1X,20A1)
0047      200 CONTINUE
      C
0048          DO 300 J=1001,1003
```

```
0049          JJ=J
0050          IS=JICVT(JJ,ID4)
0051          CALL IMAGE(ID4)
0052          ENCODE(8,301,OUT(1)) I3
0053      301 FORMAT(A8)
0054          OUT(8)=BD4(4)
0055          OUT(9)=' '
0056          DO 302 I=1,4
0057      302 OUT(I+9)=BD4(I)
      D       OUT(10)='0'
0058          OUT(14)=' '
0059          OUT(15)='3'
0060          DO 303 I=16,20
0061      303 OUT(I)=' '
0062          WRITE(1'J)(OUT(I),I=1,20)
      D       WRITE(7,304)(OUT(I),I=1,20)
0063      304 FORMAT(1X,20A1)
0064      300 CONTINUE
      C
0065          J1=1
0066          J2=1
0067          IREC=1006
0068          DO 400 JR=1,9,3
0069          ENCODE(4,401,OUT(1)) I4
0070      401 FORMAT(A4)
0071          OUT(5)=' '
0072          IS=JICVT(J1,ID4)
0073          CALL IMAGE(ID4)
0074          DO 402 J=1,4
0075      402 OUT(J+5)=BD4(J)
0076          OUT(10)=' '
0077          DO 403 J=1,3
0078          IS=JICVT(J2,ID4)
0079          CALL IMAGE(ID4)
0080          DO 404 I=1,3
0081      404 OUT(I+10)=BD4(I+1)
0082          OUT(14)=' '
0083          OUT(15)='3'
0084          DO 405 I=16,20
0085      405 OUT(I)=' '
0086          J2=J2+1
0087          WRITE(1'IREC)(OUT(I),I=1,20)
      D       WRITE(7,304)(OUT(I),I=1,20)
0088      403 CONTINUE
0089          J2=1
0090          J1=J1+1
0091      400 CONTINUE
0092          DO 500 J=1,2
0093          DO 501 J1=1,4
0094      501 OUT(J1)='0'
0095          OUT(5)="040
0096          DO 502 J1=1,4
0097      502 OUT(J1+5)='0'
0098          DO 503 J1=10,20
```

```
FORTRAN IV      V02.1-11   Mon 01-Oct-79 12:00:55              PAGE 003

0099    503  OUT(J1)="040
0100         OUT(15)='3'
0101         WRITE(1'1003+J)(OUT(I),I=1,20)
0102    500  CONTINUE
0103         OUT(1)='P'
0104         OUT(2)='R'
0105         OUT(3)='E'
0106         OUT(4)='S'
0107         OUT(5)='S'
0108         OUT(6)=' '
0109         OUT(7)='R'
0110         OUT(8)='E'
0111         OUT(9)='T'
0112         OUT(10)='U'
0113         OUT(11)='R'
0114         OUT(12)='N'
0115         DO 600 J=13,20
0116    600  OUT(J)=' '
0117         OUT(15)='1'
0118         WRITE(1'1015)(OUT(J),J=1,20)
0119         OUT(15)='2'
0120         WRITE(1'1016)(OUT(J),J=1,20)
0121         CALL CLOSE(1)
0122         STOP'BUILD COMPLETED'
0123         END
```



```
FORTRAN IV      V02.1-11   Mon 01-Oct-79 12:01:07              PAGE 001

0001         SUBROUTINE IMAGE(I14)
0002         BYTE BD4
0003         INTEGER*4 I14
0004         COMMON/IM/BD4(4)
0005         ENCODE(4,1,BD4) I14
0006      1  FORMAT(4I)
0007         DO 2 I=1,4
0008         IF(BD4(I) .LE. 57 .AND. BD4(I) .GE. 48) GOTO 2
0010         BD4(I)='0'
0011      2  CONTINUE
0012         RETURN
0013         END
```

2.4 FDMLDR LOADER PROGRAM

FDMLDR program is the Fortran loading program which is run on host processor (PDP-11/V03) which enables the user of the MSCDM to down-line load object files (.SAV) to the LSI-11 nodes, start loaded program or request status of the nodes.

2.4.1 Program Description

2.4.1.1 Program FDMLDR (FORTRAN)

This program is the driver for FDMLDR and controls the format of the CRT, opening and closing files, and the user's interactions.

2.4.1.2 Subroutine Loader (FORTRAN)

This subroutine is called by FDMLDR to load a specific node with the object file assigned to logical unit 9.

2.4.1.3 Subroutine ACKNAK (FORTRAN)

This subroutine checks the response coming back to determine whether the current packet was ACKed or NAKed.

2.4.1.4 Subroutine ISET (FORTRAN)

This subroutine is called to place the cursor of the VT52 to the line displaying the current node being loaded.

2.4.1.5 Subroutine LDRMAC (MACRO)

This subroutine contains the MACROs for the VT52 (MVTP), LIU handlers (INIT, LIO, WTLOOP).

2.4.2 Use of FDMLDR

Refer to MSCDM User's Manual for operating procedures.

```
0001          PROGRAM  FDMLDR
0002          BYTE OUTBF,RTIM(8),INBF
0003          INTEGER*2 ILID,ILND,IACK,INAK,IFLT,IGOT,FCNT
0004          REAL*8 NRFILE(16),FILE
0005          COMMON/LOOP1/OUTBF(256),INBF(256),FILE(2)
0006          COMMON/LOOP2/ILID(8),ILND(8),IGOT,IACK,INAK,IFLT
0007          DATA NRFILE/'SIGGEN.L','DA        ','NODE21.S','AV        ',
             1            'NODE22.S','AV        ','NODE23.S','AV        ',
             2            'NODE25.S','AV        ','NODE26.S','AV        ',
             3            'NODE27.S','AV        ','NODE28.S','AV        ',/
0008          ILID(1)="004
0009          ILID(2)="003
0010          ILID(3)="006
0011          ILID(4)="007
0012          ILID(5)="010
0013          ILID(6)="011
0014          ILID(7)="002
0015          ILID(8)="010
0016          ILND(1)=21
0017          ILND(2)=22
0018          ILND(3)=23
0019          ILND(4)=25
0020          ILND(5)=26
0021          ILND(6)=27
0022          ILND(7)=28
0023          ILND(8)=20
0024          ILCMD=0
0025          ISLT=0
      C*******************************
0026        1 CALL DABLE
0027          WRITE(7,2) "012,"012
0028          WRITE(7,2) "033,"110
0029          WRITE(7,2) "033,"112
0030          WRITE(7,2) "012,"012
0031          WRITE(7,2) "033,"110
0032        2 FORMAT(1X,1A1,1A1)
0033          CALL INIT(ICND,IARG1,IARG2,IARG3,IARG4)
0034          IF(ICND .EQ. 1) GOTO 4
0036          WRITE(7,3)
0037        3 FORMAT(//,1X,'ERROR DEVELOPED-INITIALIZING NODE24',/,1X
             1           ,'LOADING PROCEDURE ABORTED!!!.')
0038          WRITE(7,17) IARG1,IARG2,IARG3,IARG4
0039       17 FORMAT(//,1X,'ACRAM  ERRORS:  ',I4,/,1X,'BUFFER ERRORS:  ',I4,
             2      /,1X,'STATUS ERRORS:  ',I4,2X,O8)
0040          CALL EXIT
0041        4 CALL MVTP("040,"040)
0042          WRITE(7,5)
0043        5 FORMAT(1X,'MSCDM (LOOP 5) LOADER/RUNNING ON NODE 24:')
0044          CALL MVTP("057,"040)
0045          WRITE(7,6)
0046        6 FORMAT(1X,'NODE 21: ',/,1X,'NODE 22: ',/,1X,'NODE 23: ',/,
             2      1X,'NODE 25: ',/,1X,'NODE 26: ',/,1X,'NODE 27: ',/,
             3      1X,'NODE 28: ',/,1X,'SIG  20: ',/)
0047          CALL MVTP("044,"040)
```

```
0048         WRITE(7,8)
0049      8  FORMAT(6X,'1. NORMAL LOAD',/,6X
                4          '2. INDIVIDUAL LOAD',/,6X
                5          '3. START NODE(S)',/,6X
                6          '4. REPORT STATUS',/,6X
                7       '5. TERMINATE')
          C*******************************
0050     16  CALL RAM("005,4)
0051         DO 15 I=1,8
0052         ITRY=1
0053     90  IFLG=1
0054         IF(I .EQ. 8) IFLG=8
0056     9   OUTBF(1)=0
0057         OUTBF(2)=IFLG
0058         OUTBF(3)=3
0059         OUTBF(4)=ILID(I)
0060         DO 111 IC1=1,4
0061     111 INBF(IC1)=0
0062         CALL WTLOOP(4)
0063         DO 70 ID=1,500
0064     70  CONTINUE
0065         CALL ACKNAK(IST)
0066         DO 89 JD=1,1000
0067     89  CONTINUE
0068         IF(IST .EQ. 1) GOTO 91
0070         ITRY=ITRY+1
0071         IF(ITRY .GT. 3) GOTO 11
0073         GOTO 90
0074     91  CALL ISET(I)
0075         WRITE(7,10)
0076     10  FORMAT('$','ON -LINE READY MODE'30X)
0077         GOTO 15
0078     11  IF(IST .NE. 2) GOTO 13
0080         CALL ISET(I)
0081         WRITE(7,12)
0082     12  FORMAT('$','ON -LINE ERROR DETECTED',30X)
0083         GOTO 15
0084     13  IF(IST .NE. -1) GOTO 9
0086         CALL ISET(I)
0087         WRITE(7,14)
0088     14  FORMAT('$','OFF-LINE NO RESPONSE',30X)
0089     15  CONTINUE
          C*******************************
0090     100 CALL ISET(9)
0091         WRITE(7,106)
0092     106 FORMAT(20X,/,20X)
0093         INT=1
0094         ILCMD=ISLT
0095         CALL MVTP("042,"040)
0096         WRITE(7,101)
0097     101 FORMAT(1X,'SELECTION ? ',10X)
0098         CALL MVTP("042,"054)
0099         READ(7,102) ISLT
0100     102 FORMAT(I)
```

```
0101          CALL MVTP("041,"040)
0102          WRITE(7,107)
0103      107 FORMAT(20X)
0104          IF(ISLT .EQ. 4) GOTO 16
0106          IF(ISLT .NE. 1) GOTO 200
0108          FCNT=0
0109          DO 501 K=1,2
0110      501 FILE(K)=0
0111          DO 502 J=1,2
0112      502 FILE(J)=NRFILE(J+FCNT)
0113          CALL LOADER(8)
0114          FCNT=2
0115          DO 105 I=1,7
0116            DO 103 K=1,2
0117            FILE(K)=0
0118      103   CONTINUE
0119          DO 104 J=1,2
0120          FILE(J)=NRFILE(J+FCNT)
0121      104 CONTINUE
0122          CALL LOADER(I)
0123          FCNT=FCNT+2
0124      105 CONTINUE
0125          INT=0
0126          OUTBF(1)=0
0127          OUTBF(2)=8
0128          OUTBF(3)=3
0129          OUTBF(4)=ILID(8)
0130          CALL WTLOOP(4)
0131          GOTO 350
      C*******************************
0132      200 IF(ISLT .NE. 2) GOTO 300
0134      201 CALL ISET(9)
0135          WRITE(7,202)
0136      202 FORMAT(1X,'NODE[  ]',10X,/,1X,'FILE[            ]')
0137          DO 203 I=1,2
0138      203   FILE(I)=0
0139          CALL MVTP("052,"045)
0140          READ(7,206) INT
0141      206 FORMAT(I)
0142          IF(INT .EQ. 99) GOTO 100
0144          CALL MVTP("054,"045)
0145          READ(7,215) FILE
0146      215 FORMAT(2A8)
0147          DO 208 I=1,8
0148          IF(I .EQ. 8 .AND. INT .EQ. 0) GOTO 208
0150           IF(INT .EQ. 0) GOTO 209
0152           IF(INT .EQ. ILND(I)) GOTO 209
0154           GOTO 208
0155      209  CALL LOADER(I)
0156      208 CONTINUE
0157          IF(INT .NE. 0) GOTO 201
0159          GOTO 100
      C*******************************
0160      300 IF(ISLT .NE. 3) GOTO 400
```

```
0162     301 CALL ISET(9)
0163         WRITE(7,302)
0164     302 FORMAT(1X,'START NODE[  ]',/,25X)
0165         CALL MVTP("052,"053)
0166         READ(7,304) INT
0167     304 FORMAT(I)
0168         IF(INT .EQ. 99) GOTO 100
0170     350 DO 305 I=1,8
0171          IF(INT .EQ. 0) GOTO 306
0173          IF(INT .EQ. ILND(I)) GOTO 306
0175          GOTO 305
0176     306 IF(I .NE. 8) GOTO 308
0178         CALL ISET(8)
0179         WRITE(7,309)
0180     309 FORMAT('$','AUTO-START AFTER LOAD',20X)
0181         GOTO 305
0182     308       CALL ISET(I)
0183          WRITE(7,307)
0184     307  FORMAT('$','ATTEMPTING TO START',30X)
0185          OUTBF(1)=0
0186          OUTBF(2)=5
0187          OUTBF(3)=3
0188          OUTBF(4)=ILID(I)
0189          CALL WTLOOP(4)
0190          DO 340 JL=1,5000
0191     340  CONTINUE
0192     305 CONTINUE
0193         IF(ISLT .EQ. 1) GOTO 450
0195         IF(INT .EQ. 0) GOTO 100
0197         GOTO 301
      C*******************************
0198     400 IF(ISLT .NE. 5) GOTO 100
0200         CALL DABLE
0201         CALL EXIT
0202     450 WRITE(7,451) "033,"110
0203     451 FORMAT(1X,1A1,1A1)
0204         WRITE(7,451) "033,"112
0205         WRITE(7,451) "033,"110
0206         CALL TIME(RTIM)
0207         WRITE(7,452) (RTIM(I),I=1,8)
0208     452 FORMAT(///,1X,'MSCDM LOOP 5 LOADED AT :  ',8A1)
0209         CALL DABLE
0210         CALL EXIT
0211         END
      C$$$$$$
```

```
0001          SUBROUTINE LOADER(INODE)
0002          BYTE OUTBF,INBF,BLOCK(4,128),RTIM(8)
0003          INTEGER*2 ILID,ILND,IGOT,IACK,INAK,IFLT
0004          REAL*8 FILE
0005          COMMON/LOOP1/OUTBF(256),INBF(256),FILE(2)
0006          COMMON/LOOP2/ILID(8),ILND(8),IGOT,IACK,INAK,IFLT
      C
0007        302 CALL ASSIGN(9,FILE)
0008          DEFINE FILE 9(300,256,U,IREC)
0009          CALL ISET(INODE)
0010          WRITE(7,3) FILE
0011        3   FORMAT('$','LOADING- ',2A8,25X)
0012          CALL ISET(INODE)
0013          IRLCT=1
0014          IFLG=6
0015          IF(INODE .EQ. 8) IFLG=9
0017        4   OUTBF(1)=0
0018          OUTBF(2)=IFLG
0019          OUTBF(3)=3
0020          OUTBF(4)=ILID(INODE)
0021          CALL WTLOOP(4)
0022          IREC=1
0023          IPC=0
0024          IECNT=0
      C
0025        100 READ(9'IREC,END=200,ERR=300) ((BLOCK(I,J),J=1,128),I=1,4)
0026          FIND(9'IREC)
0027          IDIR=0
0028          DO 101 I=1,4
0029          IPC=IPC+1
0030          IF(IPC .EQ. 100) IPC=1
0032          IECNT=0
0033           DO 102 J=1,128
0034           OUTBF(J+2)=BLOCK(I,J)
0035        102  CONTINUE
0036          IFLG=0
0037          IF(INODE .EQ. 8) IFLG=7
0039          OUTBF(1)=IPC
0040          OUTBF(2)=IFLG
0041          OUTBF(131)="003
0042          OUTBF(132)=ILID(INODE)
0043          DO 110 IC=1,4
0044        110 INBF(IC)=0
0045        103 CALL WTLOOP(132)
0046          IF(INODE .NE. 8) GOTO 104
0048          DO 121 ID1=1,9000
0049        121 CONTINUE
0050        104 CALL ACKNAK(ISTAT)
0051        105 IF(ISTAT .EQ. 1) GOTO 101
0053          WRITE(7,107) IPC,ISTAT
0054        107 FORMAT('$','RESENDING PACKET ',I4,2X,'ERROR: ',I2,25X)
0055          CALL ISET(INODE)
0056          IECNT=IECNT+1
0057          IF(IECNT .LT. 4) GOTO 103
```

```
0059          WRITE(7,108) IRLCT
0060      108 FORMAT('$','RELOADING ATTEMPT# ',I2,29X)
0061          CALL ISET(INODE)
0062          IRLCT=IRLCT+1
0063          IF(IRLCT .LT. 4) GOTO 4
0065          WRITE(7,109)
0066      109 FORMAT('$','NODE FAILED TO LOAD!!!',28X)
0067          CALL ISET(INODE)
0068          CALL CLOSE(9)
0069          RETURN
      C
0070      101 CONTINUE
0071          GOTO 100
0072      200 CALL TIME(RTIM)
0073          CALL ISET(INODE)
0074          WRITE(7,201) (RTIM(J),J=1,8),FILE
0075      201 FORMAT('$','LOADING COMPLETED ',8A1,'  ',2A8,4X)
0076          CALL CLOSE(9)
0077          RETURN
0078      300 IF(IDIR .EQ. 1) GOTO 301
0080          CALL CLOSE(9)
0081          IDIR=1
0082          CALL ISET(INODE)
0083          WRITE(7,304) FILE
0084      304 FORMAT('$','INSERT DISK WITH --> ',2A8,12X)
0085          READ(7,305) IDUM
0086      305 FORMAT(A)
0087          CALL ASSIGN(1,'DX0:DUMMY.MAC')
0088          CALL CLOSE(1)
0089          GOTO 302
0090      301 WRITE(7,306) FILE
0091      306 FORMAT('$','ERROR FILE NOT FOUND- ',2A8,12X)
0092          CALL CLOSE(9)
0093          RETURN
0094          END
      C$$$$$$
```

```
0001          SUBROUTINE ACKNAK(IRTN)
0002          INTEGER*2 ILID,ILND,IGOT,IACK,INAK,IFLT,ICNT
0003          COMMON/LOOP2/ILID(8),ILND(8),IGOT,IACK,INAK,IFLT
      C
0004          ICNT=0
0005      100 DO 1,I=1,600
0006          IF(IGOT .EQ. 1) GOTO 2
0008      1   CONTINUE
0009      2   ICNT=ICNT+1
0010          IF(IACK .NE. 1) GOTO 3
0012          IRTN=1
0013          GOTO 999
0014      3   IF(INAK .NE. 1) GOTO 4
0016          IRTN=2
0017          GOTO 999
0018      4   IF(IFLT .NE. 1) GOTO 5
0020          IRTN=-2
0021          GOTO 999
0022      5   IF(ICNT .LT. 5) GOTO 100
0024          IRTN=-1
0025          GOTO 999
0026    999   IACK=0
0027          INAK=0
0028          IFLT=0
0029          IGOT=0
0030          RETURN
0031          END
      C$$$$$$
```

```
FORTRAN IV      V02.1-11   Mon 01-Oct-79 12:02:11              PAGE 001

0001          SUBROUTINE ISET(ILN)
0002      1   IF(ILN .NE. 1) GOTO 2
0004          WRITE(7,100) "033,"110
0005          CALL MVTP("056,"052)
0006          GOTO 999
0007      2   IF(ILN .NE. 2) GOTO 3
0009          WRITE(7,100) "033,"110
0010          CALL MVTP("057,"052)
0011          GOTO 999
0012      3   IF(ILN .NE. 3) GOTO 4
0014          WRITE(7,100) "033,"110
0015          CALL MVTP("060,"052)
0016          GOTO 999
0017      4   IF(ILN .NE. 4) GOTO 5
0019          WRITE(7,100) "033,"110
0020          CALL MVTP("061,"052)
0021          GOTO 999
0022      5   IF(ILN .NE. 5) GOTO 6
0024          WRITE(7,100) "033,"110
0025          CALL MVTP("062,"052)
0026          GOTO 999
0027    . 6   IF(ILN .NE. 6) GOTO 7
0029          WRITE(7,100) "033,"110
0030          CALL MVTP("063,"052)
0031          GOTO 999
0032      7   IF(ILN .NE. 7) GOTO 8
0034          WRITE(7,100) "033,"110
0035          CALL MVTP("064,"052)
0036          GOTO 999
0037      8   IF(ILN .NE. 8) GOTO 9
0039          WRITE(7,100) "033,"110
0040          CALL MVTP("065,"052)
0041          GOTO 999
0042      9   IF(ILN .NE. 9) GOTO 999
0044          WRITE(7,100) "033,"110
0045          CALL MVTP("052,"040)
0046          GOTO 999
0047      100 FORMAT(1X,1A1,1A1)
0048      999 RETURN
0049          END
```

```
                                        .TITLE LDRMAC.MACRO
                                        .IDENT  /V1.0/
                                        .GLOBL  MVTP,WTLOOP,INIT,RAM,DABLE
000000                                  .PSECT

                172410                  BAR=    172410
                172412                  WCR=    172412
                172414                  CSR=    172414
                172416                  IOBUF=  172416
                172416                  OPREG=  172416
                177560                  RCSR=   177560
                177562                  RBUF=   177562
                177564                  XCSR=   177564
                177566                  XBUF=   177566
                000002                  ARG1=   2
                000004                  ARG2=   4
                000006                  ARG3=   6
                000010                  ARG4=   10
                000012                  ARG5=   12
                100000                  BIT15=  100000
                040000                  BIT14=  40000
                000400                  BIT08=  400
                000200                  BIT07=  200
                000100                  BIT06=  100
                000040                  BIT05=  40
                000020                  BIT04=  20
                000010                  BIT03=  10
                000004                  BIT02=  4
                000002                  BIT01=  2
                000001                  BIT00=  1

                                        ;VT52 HANDLER

000000  105737  177564          MVTP:   TSTB    @#XCSR
000004  100375                          BPL     .-4
000006  012737  000033  177566          MOV     #033, @#XBUF
000014  105737  177564                  TSTB    @#XCSR
000020  100375                          BPL     .-4
000022  012737  000131  177566          MOV     #'Y, @#XBUF
000030  105737  177564                  TSTB    @#XCSR
000034  100375                          BPL     .-4
000036  017537  000002  177566          MOV     @ARG1(R5), @#XBUF
000044  105737  177564                  TSTB    @#XCSR
000050  100375                          BPL     .-4
000052  017537  000004  177566          MOV     @ARG2(R5), @#XBUF
000060  000240                          NOP
000062  000207                          RTS     PC
```

```
                                        ;INITIALIZE LIU
000064  012737  010420  172416  INIT:   MOV     #4368., @#OPREG         ;MOD STAT
000072  012737  004414  172416          MOV     #4414,  @#OPREG         ;SET SWITCHES
000100  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000104  100375                         BPL     .-4                     ;LOOP UNTIL
000106  105037  172414                  CLRB    @#CSR
000112  012737  004540  172416          MOV     #4540,  @#OPREG         ;RESET SWITCHS
000120  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000124  100375                         BPL     .-4                     ;LOOP UNTIL
000126  105037  172414                  CLRB    @#CSR
000132  012700  000124                  MOV     #124, R0                ;INTERRUPT VECTOR
000136  012720  001026'                 MOV     #LIO, (R0)+             ;ADDRESS OF LIO
000142  012710  000340                  MOV     #340, (R0)              ;PRI =340
000146  005003                         CLR     R3                      ;CLEAR ERROR COUNT
000150  005067  001622                  CLR     ERAM
000154  005067  001620                  CLR     EBUF
000160  005067  001616                  CLR     ESTAT

                                        ;INITIALIZE ACRAM

000164  005004                 LINIT:  CLR     R4                      ;CLEAR LOOP COUNT
000166  052737  000001  172414          BIS     #BIT00, @#CSR           ;INIT LIU
000174  000240                         NOP
000176  000240                         NOP
000200  042737  000001  172414          BIC     #BIT00, @#CSR           ;CLEAR LIU
000206  012737  010402  172416          MOV     #4354., @#OPREG         ;LDADR
000214  012737  004400  172416          MOV     #2304., @#OPREG         ;ADDRESS=0
000222  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000226  100375                         BPL     .-4                     ;NO RETRY
000230  105037  172414                  CLRB    @#CSR
000234  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
000242  012700  000400                  MOV     #256., R0               ;COUNTER
000246  012737  004407  172416  1$:     MOV     #2311., @#OPREG         ;WRITE A NULL
000254  105737  172414                  TSTB    @#CSR                   ;GOOD WD
000260  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
000262  105037  172414                  CLRB    @#CSR
000266  077011                         SOB     R0,     1$              ;LOOP UNTIL ZERO
000270  012700  000400                  MOV     #256.,  R0              ;COUNTER
000274  012737  001400  172416  2$:     MOV     #768.,  @#OPREG         ;RD
000302  105737  172414                  TSTB    @#CSR                   ;GOOD RD
000306  100375                         BPL     .-4                     ;NO RETRY
000310  105037  172414                  CLRB    @#CSR
000314  013701  172416                  MOV     @#IOBUF, R1             ;FETCH DATA
000320  042701  177760                  BIC     #177760, R1             ;CLEAR BITS
000324  022701  000007                  CMP     #7,      R1             ;DATA=NULL
000330  001403                         BEQ     3$                      ;YES
000332  005204                         INC     R4                      ;NO REPORT IT
000334  005267  001436                  INC     ERAM                    ;RAM ERROR
000340  077023                 3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO

                                        ;INITIALIZE BUFFERS

000342  012701  010410          INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
000346  012700  010440                  MOV     #10440, R0              ;SEL INBUF0
```

```
 58 000352  004767  000106                    JSR     PC,     ZEROBP          ;SET POINTER=0
 59 000356  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
 60 000364  105737  172414                    TSTB    @#CSR                   ;GOOD READ
 61 000370  100375                            BPL     .-4                     ;NO RETRY
 62 000372  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 63 000376  012701  010610          INBUF1:   MOV     #10610, R1              ;RDBUFADR COMMAND
 64 000402  012700  010640                    MOV     #10640, R0              ;SEL INBUF1 COMMAND
 65 000406  004767  000052                    JSR     PC,     ZEROBP          ;POINTER=0
 66 000412  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
 67 000420  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 68 000424  100375                            BPL     .-4                     ;NO RETRY
 69 000426  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 70 000432  012701  010510          OTBUF0:   MOV     #10510, R1              ;RDBUFADR COMMAND
 71 000436  012700  010540                    MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
 72 000442  004767  000016                    JSR     PC,     ZEROBP
 73 000446  012701  010710          OTBUF1:   MOV     #10710, R1              ;RDBUFADR COMMAND
 74 000452  012700  010740                    MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
 75 000456  004767  000002                    JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
 76 000462  000464                            BR      STAT                    ;GO CLEAR STATUS
 77 000464  010137  172416          ZEROBP:   MOV     R1,     @#OPREG         ;RDBUFADR
 78 000470  012737  001400  172416            MOV     #1400,  @#OPREG         ;RD
 79 000476  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 80 000502  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
 81 000504  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 82 000510  013702  172416                    MOV     @#IOBUF, R2             ;FETCH POINTER
 83 000514  042702  177400                    BIC     #177400, R2             ;CLEAR MST BYTE
 84 000520  010037  172416                    MOV     R0,     @#OPREG         ;SEL BUFFER
 85 000524  022702  000000          4$:       CMP     #0,     R2              ;POINTER=0
 86 000530  001412                            BEQ     5$
 87 000532  012737  001400  172416            MOV     #1400,  @#OPREG         ;NO FALSE RD
 88 000540  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 89 000544  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
 90 000546  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 91 000552  005302                            DEC     R2                      ;POINTER-1
 92 000554  000763                            BR      4$
 93 000556  010137  172416          5$:       MOV     R1,     @#OPREG         ;RDBUFADR
 94 000562  012737  001400  172416            MOV     #1400,  @#OPREG         ;RD
 95 000570  105737  172414                    TSTB    @#CSR                   ;GOOD RD
 96 000574  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
 97 000576  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
 98 000602  013702  172416                    MOV     @#IOBUF, R2             ;FETCH NEW POINTER
 99 000606  042702  177400                    BIC     #177400, R2             ;CLEAR MST BYTE
100 000612  022702  000000                    CMP     #0,     R2              ;POINTER=0
101 000616  001403                            BEQ     6$
102 000620  005204                            INC     R4                      ;REPORT ERROR
103 000622  005267  001152                    INC     EBUF                    ;BUFFER ERROR
104 000626  010037  172416          6$:       MOV     R0,     @#OPREG         ;SEL BUFFER
105 000632  000207                            RTS     PC                      ;RETURN
106
107                                           ;INITIALIZE STATUS
108
109 000634  012737  010400  172416  STAT:     MOV     #4352., @#OPREG         ;WCR:RS
110 000642  012737  002400  172416            MOV     #1280., @#OPREG         ;RS(FALSE)
111 000650  012737  002400  172416            MOV     #1280., @#OPREG         ;RS
112 000656  013700  172416                    MOV     @#OPREG, R0             ;FETCH STATUS
113 000662  042700  177400                    BIC     #177400, R0             ;CLEAR MST BYTE
114 000666  022700  000000                    CMP     #0,     R0              ;STATUS=0 ?
```

```
115 000672  001403                          BEQ     7$
116 000674  005204                          INC     R4                          ;STATUS BAD REPORT
117 000676  005267  001100                  INC     ESTAT                       ;STATUS ERROR
118 000702  022704  000000          7$:     CMP     #0,     R4                  ;WERE THERE ERRORS
119 000706  001406                          BEQ     8$                          ;NO GOOD INIT
120 000710  005203                          INC     R3                          ;ERROR +1
121 000712  022703  000005                  CMP     #5,     R3                  ;ERROR LIMIT YET
122 000716  001421                          BEQ     9$                          ;YES RETURN A 1(ERROR)
123 000720  000167  177240                  JMP     LINIT                       ;NO RESTART VINIT
124 000724  012775  000001  000002  8$:     MOV     #1,     @ARG1(R5)           ;GOOD INIT
125 000732  005075  000004                  CLR     @ARG2(R5)
126 000736  005075  000006                  CLR     @ARG3(R5)
127 000742  005075  000010                  CLR     @ARG4(R5)
128 000746  005075  000012                  CLR     @ARG5(R5)
129 000752  052737  040000  172414          BIS     #BIT14, @#CSR               ;ENABLE INTERRUPTS
130 000760  000207                          RTS     PC
131 000762  012775  177777  000002  9$:     MOV     #-1,    @ARG1(R5)           ;BAD INIT
132 000770  016775  001002  000004          MOV     ERAM,   @ARG2(R5)
133 000776  016775  000776  000006          MOV     EBUF,   @ARG3(R5)
134 001004  016775  000772  000010          MOV     ESTAT,  @ARG4(R5)
135 001012  010075  000012                  MOV     R0,     @ARG5(R5)
136 001016  042737  040000  172414          BIC     #BIT14, @#CSR               ;DISABLE LIU
137 001024  000207                          RTS     PC
138                                         .ENABLE LSB
```

```
 1
 2                                          ;LIU HANDLER
 3
 4 001026  010046                  LIO:    MOV     R0,     -(SP)           ;SAVE REGISTERS
 5 001030  010146                          MOV     R1,     -(SP)
 6 001032  010246                          MOV     R2,     -(SP)
 7 001034  010346                          MOV     R3,     -(SP)
 8 001036  010446                          MOV     R4,     -(SP)
 9 001040  010546                          MOV     R5,     -(SP)
10 001042  005067  000726                  CLR     DATA                    ;CLEAR VAR.
11 001046  005737  172414                  TST     @#CSR                   ;INT
12 001052  100077                          BPL     RTI$                    ;NO RETURN
13 001054  042737  040000  172414          BIC     #BIT14, @#CSR           ;DISABLE
14 001062  012737  010400  172416          MOV     #4352., @#OPREG         ;WCR : RS(0)
15 001070  012737  002400  172416          MOV     #1280., @#OPREG         ;RD
16 001076  013701  172416                  MOV     @#IOBUF, R1
17 001102  042701  177400                  BIC     #177400, R1             ;CLEAR UNUSED BITS
18
19                                         ;BUFFER OVERFLOW
20
21 001106  132701  000004          B2$:    BITB    #BIT02, R1              ;OV-FL
22 001112  001406                          BEQ     B3$
23 001114  012702  010410                  MOV     #4360., R2              ;COM=SEL INBUF 0
24 001120  012703  010440                  MOV     #4384., R3              ;COM=RDBUFPNT
25 001124  004767  000146                  JSR     PC, EMBF                ;EMPTY BUFFER
26 001130  132701  000010          B3$:    BITB    #BIT03, R1              ;OV-FL
27 001134  001406                          BEQ     B0$
28 001136  012702  010610                  MOV     #4488., R2              ;COM=SEL INBUF 1
29 001142  012703  010640                  MOV     #4512., R3              ;COM=RDBUFPNT
30 001146  004767  000124                  JSR     PC, EMBF
31
32                                         ;BUFFERS FULL
33
34 001152  132701  000001          B0$:    BITB    #BIT00, R1              ;BUFFER 0 FULL
35 001156  001415                          BEQ     B1$                     ;NO CK 1
36 001160  012702  010410                  MOV     #4360., R2              ;YES, COM=RDBUFADR
37 001164  012703  010440                  MOV     #4384., R3              ;COM=SEL INBUF0
38 001170  012704  000001                  MOV     #BIT00, R4              ;WHICH CRC BIT
39 001174  004767  000076                  JSR     PC, EMBF                ;GO EMPTY AND LOAD
40 001200  005767  000570                  TST     DATA
41 001204  100402                          BMI     B1$
42 001206  004767  000264                  JSR     PC, PARSE               ;GO CHECK PACKET
43 001212  132701  000002          B1$:    BITB    #BIT01, R1              ;BUFFER 1 FULL
44 001216  001415                          BEQ     RTI$                    ;NO LOOP BACK
45 001220  012702  010610                  MOV     #4488., R2              ;YES,COM=RDBUFADR
46 001224  012703  010640                  MOV     #4512., R3              ;COM=SEL INBUF1
47 001230  012704  000002                  MOV     #BIT01, R4              ;WHICH CRC BIT
48 001234  004767  000036                  JSR     PC,     EMBF            ;EMPTY AND LOAD
49 001240  005767  000530                  TST     DATA
50 001244  100402                          BMI     RTI$
51 001246  004767  000224                  JSR     PC, PARSE               ;GO CHECK PACKET
52
53 001252  012605                 RTI$:   MOV     (SP)+, R5               ;RESTORE REGISTERS
54 001254  012604                          MOV     (SP)+, R4
55 001256  012603                          MOV     (SP)+, R3
56 001260  012602                          MOV     (SP)+, R2
57 001262  012601                          MOV     (SP)+, R1
```

```
 58 001264  012600                          MOV     (SP)+, R0
 59 001266  052737  040000  172414          BIS     #BIT14, @#CSR
 60 001274  000002                          RTI
 61
 62                                         ;EMPTY BUFFERS
 63
 64 001276  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG         ;RS (CRC)
 65 001304  012737  002400  172416          MOV     #1280., @#OPREG         ;RS CMD
 66 001312  013705  172416                  MOV     @#IOBUF, R5             ;FETCH STATUS
 67 001316  130405                          BITB    R4, R5                  ;CRC OK
 68 001320  001003                          BNE     5$                      ;YES CONTINUE
 69 001322  012767  177777  000444          MOV     #-1,    DATA            ;FLAG BAD CRC
 70 001330  012737  000400' 172410  5$:     MOV     #INBF, @#BAR            ;ADDRESS TO WRITE
 71 001336  010237  172416                  MOV     R2,     @#OPREG         ;RDBUFADR
 72 001342  012737  001400  172416          MOV     #768.,  @#OPREG         ;READ POINTER
 73 001350  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 74 001354  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
 75 001356  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 76 001362  013704  172416                  MOV     @#IOBUF, R4             ;READ POINTER
 77 001366  042704  177400                  BIC     #177400, R4             ;CLEAR UNUSED BITS
 78 001372  005404                  6$:     NEG     R4                      ;2'S COMP
 79 001374  010437  172412                  MOV     R4,  @#WCR              ;BYTE COUNT
 80 001400  010337  172416                  MOV     R3,     @#OPREG         ;SEL BUFFER
 81 001404  012737  001400  172416          MOV     #768.,  @#OPREG         ;FALSE RD
 82 001412  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 83 001416  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
 84 001420  105037  172414                  CLRB    @#CSR
 85 001424  012737  021000  172416          MOV     #8704., @#OPREG         ;FIRE DMA RD
 86 001432  000240                          NOP                             ;DELAY
 87 001434  105737  172414                  TSTB    @#CSR                   ;GOOD DMA
 88 001440  100403                          BMI     7$                      ;YES
 89 001442  012767  177777  000324          MOV     #-1,  DATA              ;BAD DMA XFER
 90 001450  105037  172414          7$:     CLRB    @#CSR                   ;CLEAR CSR
 91 001454  012737  004400  172416          MOV     #2304., @#OPREG         ;RESET POINTER TO 255
 92 001462  105737  172414                  TSTB    @#CSR                   ;GOOD WD
 93 001466  100375                          BPL     .-4                     ;NO RETRY
 94 001470  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 95 001474  000207                  8$:     RTS     PC
 96                                         .ENABLE LSB
 97
 98 001476  012702  000400'          PARSE:  MOV     #INBF, R2               ;ADDRESS OF PACKET
 99 001502  116203  000001                  MOVB    1(R2), R3               ;FETCH COMMAND
100 001506  122703  000002          1$:     CMPB    #2,     R3              ;=ACK
101 001512  001007                          BNE     2$
102 001514  012767  000001  000040'         MOV     #1,     IGOT            ;YES REPORT
103 001522  012767  000001  000042'         MOV     #1,     IACK
104 001530  000207                          RTS     PC                      ;RETURN
105 001532  122703  000003          2$:     CMPB    #3,     R3              ;=NAK
106 001536  001007                          BNE     3$
107 001540  012767  000001  000040'         MOV     #1,     IGOT            ;YES REPORT
108 001546  012767  000001  000044'         MOV     #1,     INAK            ;FLAG
109 001554  000207                          RTS     PC
110 001556  012767  000001  000046' 3$:     MOV     #1,     IFLT
111 001564  012767  000001  000040'         MOV     #1,     IGOT
112 001572  000207                          RTS     PC
113
```

```
                                        ;WRITE RAM
001574  017500  000002          RAM:    MOV     @ARG1(R5), R0           ;ADDRESS
001600  017501  000004                  MOV     @ARG2(R5), R1           ;WRITE DATA
001604  012737  010402  172416          MOV     #4354., @#OPREG         ;SEL LDACTR
001612  062700  004400                  ADD     #2304., R0              ;WD/DATA (ADDR)
001616  010037  172416                  MOV     R0,     @#OPREG         ;WRITE DATA
001622  105737  172414                  TSTB    @#CSR                   ;VALID WRITE
001626  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
001630  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
001634  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
001642  062701  004400                  ADD     #2304., R1              ;WD/DATA (CMD)
001646  010137  172416                  MOV     R1,     @#OPREG         ;WRITE
001652  105737  172414                  TSTB    @#CSR                   ;VALID WRITE
001656  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
001660  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
001664  000207                         RTS     PC
                                        ;DISABLE INTERRUPTS
001666  042737  040000  172414  DABLE:  BIC     #BIT14, @#CSR           ;DIABLE LIU
001674  000207                         RTS     PC
                                        ;WRITE LOOP
001676  017502  000002          WTLOOP: MOV     @ARG1(R5), R2           ;BYTE COUNT
001702  012737  000000' 172410          MOV     #OUTBF, @#BAR           ;BUS ADDRESS
001710  005402                         NEG     R2                      ; 2'S COMP COUNT
001712  010237  172412                  MOV     R2,     @#WCR           ;COUNT
001716  012737  010540  172416          MOV     #4448., @#OPREG         ;OB0 COMMAND
001724  012737  024000  172416          MOV     #10240., @#OPREG        ;DMA GO
001732  000240                         NOP                             ;INTERFACE TIME
001734  105737  172414                  TSTB    @#CSR                   ;DMA OK
001740  100401                         BMI     CMD                     ;YES
001742  000000                         HALT
001744  012737  010420  172416  CMD:    MOV     #4368., @#OPREG         ;MODSTAT COMMAND
001752  012737  004421  172416          MOV     #4421,  @#OPREG         ;BUFFERS FULL
001760  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
001764  100375                         BPL     .-4                     ;NO LOOP UNTIL
001766  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
001772  000207                         RTS     PC

001774                                  .PSECT

001774  000000          DATA:   .WORD   0
001776          ERAM:   .BLKW
002000          EBUF:   .BLKW
002002          ESTAT:  .BLKW

000000                                  .PSECT  LOOP1,RW,D,GBL,REL,OVR
000000          OUTBF:  .BLKB   256.
000400          INBF:   .BLKB   256.
001000          FILE:   .BLKW   4.

000000                                  .PSECT  LOOP2,RW,D,GBL,REL,OVR
```

```
     58 000000                          ILID:   .BLKW   8.
     59 000020                          ILND:   .BLKW   8.
     60 000040                          IGOT:   .BLKW
     61 000042                          IACK:   .BLKW
     62 000044                          INAK:   .BLKW
     63 000046                          IFLT:   .BLKW
     64
     65
     66          000001                         .END
```

SYMBOL TABLE

```
ARG1  = 000002          BIT06 = 000100          DATA    001774R         INBF    000400R     002 PARSE   001476R
ARG2  = 000004          BIT07 = 000200          EBUF    002000R         INBUF0  000342R         RAM     001574RG
ARG3  = 000006          BIT08 = 000400          EMBF    001276R         INBUF1  000376R         RBUF  = 177562
ARG4  = 000010          BIT14 = 040000          ERAM    001776R         INIT    000064RG        RCSR  = 177560
ARG5  = 000012          BIT15 = 100000          ESTAT   002002R         IOBUF = 172416          RTI$    001252R
BAR   = 172410          B0$     001152R         FILE    001000R     002 LINIT   000164R         STAT    000634R
BIT00 = 000001          B1$     001212R         IACK    000042R     003 LIO     001026R         WCR   = 172412
BIT01 = 000002          B2$     001106R         IFLT    000046R     003 MVTP    000000RG        WTLOOP  001676RG
BIT02 = 000004          B3$     001130R         IGOT    000040R     003 OPREG = 172416          XBUF  = 177566
BIT03 = 000010          CMD     001744R         ILID    000000R     003 OTBUF0  000432R         XCSR  = 177564
BIT04 = 000020          CSR   = 172414          ILND    000020R     003 OTBUF1  000446R         ZEROBP  000464R
BIT05 = 000040          DABLE   001666RG        INAK    000044R     003 OUTBF   000000R     002

. ABS.  000000     000
        002004     001
LOOP1   001010     002
LOOP2   000050     003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:LDRMAC=DK:LDRMAC
```

```
ASS DX0: DK .
R LINK
DX1:FDMLDR=DX1:FDMLDR/C
DX1:LDRMAC//

ASS DX1: DK
```

3.1 DIAGNOSTICS

3.1.1 LSI-11 CPU TEST

Source File: LSICPU.MAC

Task File: LSICPU.SAV

PURPOSE: To test LSI-11/2 CPU

DESCRIPTION: This program tests all LSI-11/2 single, double and jump instruction sets. The program loops until an error is detected in the CPU; when detected, the current program address is displayed showing what instruction set is causing the error.

```
                                        .TITLE LSI-11 CPU-DIAGNOSTICS
                                        .IDENT  /V1.0/
000000                                  .ASECT
        001000                          .=1000

                                ;*****************************************
                                ;*                                       *
                                ;*  LSI-11  LIAGNOSTICS                  *
                                ;*                                       *
                                ;*      TEST(1)- SINGLE OPERAND INSTR.   *
                                ;*          (2)- BYTE SINGLE OPERAND     *
                                ;*          (3)- DOUBLE OPERAND INSTR.   *
                                ;*          (4)- JUMP INSTRUCTIONS       *
                                ;*          (5)- TST & TSTB INSTR.       *
                                ;*          (6)- BYTE DOUBLE OPERAND     *
                                ;*          (7)- CPU GENERAL             *
                                ;*          (8)- BYTE GENERAL            *
                                ;*  NOTE: PROGRAM HALTS AT (MSTART) ON   *
                                ;*          A DETECTED ERROR USE (SP)    *
                                ;*          TO FIND ERROR IN CPU.        *
                                ;*                                       *
                                ;*****************************************

001000  000005          CLEAR:  RESET                                   ;RESET BUS
001002  012700  000340          MOV     #340,   R0                      ;PRI=7
001006  106400                  MTPS    R0                              ;SET PS
001010  012706  000500          MOV     #500,   SP                      ;INITIALIZE STACK
001014  012702  000400          MOV     #400,   R2                      ;R2=TOP OF VECTOR AREA
001020  005042          CLR$:   CLR     -(R2)                           ;INSERT A HALT
Z 001022  010242                MOV     R2,     -(R2)                   ;POINT VECTOR TO HALT
001024  005702                  TST     R2                              ;?R2=0 (DONE)
001026  001374                  BNE     CLR$                            ;NO CONTINUE
001030  012737  001000  000040  MOV     #CLEAR, @#40                    ;RESTART ADDRESS
001036  012737  001046  000100  MOV     #TIME,  @#100                   ;CLOCK RETURN
001044  000401                  BR      TEST1                           ;BEGIN
001046  000002          TIME:   RTI                                     ;IF CLOCK ON RETURN


                                ;*********************************
                                ;* TEST1:
                                ;* CHECK ALL SINGLE OPERAND
                                ;* INSTRUCTIONS WITH MODE ZERO
                                ;*********************************
                                                        ; REG       NZVC
001050  005001          TEST1:  CLR     R1              ;000000     0100
001052  005101                  COM     R1              ;177777     1001
001054  005401                  NEG     R1              ;000001     0001
001056  006001                  ROR     R1              ;100000     1000
001060  005501                  ADC     R1              ;100001     1001
001062  000261                  SEC                     ;100001     1001
001064  006101                  ROL     R1              ;000003     0011
001066  006201                  ASR     R1              ;000001     0011
001070  005601                  SBC     R1              ;000000     0100
001072  006301                  ASL     R1              ;000000     0100
001074  005201                  INC     R1              ;000001     0000
001076  005301                  DEC     R1              ;000000     0100
001100  001076                  BNE     MSTART                          ;ERROR
```

```
                                        ;**********************************
                                        ;* TEST2:
                                        ;*CHECK ALL BYTE SINGLE OPERAND
                                        ;*INSTRUCTIONS WITH MODE ZERO
                                        ;**********************************

001102  112702  000377        TEST2:  MOVB    #000377,R2      ;177777    1000
001106  105002                  CLRB    R2              ;177400    0100
001110  105102                  COMB    R2              ;177777    1001
001112  105602                  SBCB    R2              ;177776    1000
001114  105202                  INCB    R2              ;177777    1000
001116  105402                  NEGB    R2              ;177401    0001
001120  105302                  DECB    R2              ;177400    0101
001122  000302                  SWAB    R2              ;000377    1000
001124  106302                  ASLB    R2              ;000376    1001
001126  106002                  RORB    R2              ;000377    1010
001130  106202                  ASRB    R2              ;000377    1001
001132  106102                  ROLB    R2              ;000377    1001
001134  105502                  ADCB    R2              ;000000    0101
001136  105702                  TSTB    R2              ;000000    0100
001140  001056                  BNE     MSTART                                  ;ERROR


                                        ;**********************************
                                        ;* TEST3:
                                        ;* CHECK ALL SOURCE MODES FOR
                                        ;* DOUBLE OPERAND WORD INSTRUCTIONS
                                        ;* WITH MODE ZERO
                                        ;**********************************

001142  012702  001324        TEST3:  MOV     #T3DATA,R2
001146  011201                  MOV     (R2),R1
001150  022201                  CMP     (R2)+,R1
001152  001051                  BNE     MSTART                                  ;ERROR
001154  063201                  ADD     @(R2)+,R1
001156  165201                  SUB     @-(R2),R1
001160  044201                  BIC     -(R2),R1
001162  056201  000004          BIS     4(R2),R1
001166  037201  000006          BIT     @6(R2),R1
001172  001441                  BEQ     MSTART                                  ;ERROR
001174  074101                  XOR     R1,R1
001176  001037                  BNE     MSTART                                  ;ERROR
```

```
                                        ;**********************************
                                        ;* TEST4:
                                        ;* CHECK ALL JUMP INSTRUCTIONS
                                        ;**********************************

001200  012701  001210          TEST4:  MOV     #JMP1,R1
001204  000111                          JMP     (R1)
001206  000433                          BR      MSTART                          ;ERROR
001210  022121          JMP1:   CMP     (R1)+,(R1)+
001212  000131                          JMP     @(R1)+
001214  001216                          .WORD   JMP3
001216  000161  000006          JMP3:   JMP     6(R1)
001222  000425                          BR      MSTART                          ;ERROR



                                        ;**********************************
                                        ;* TEST 5:
                                        ;* TEST ALL TST & TSTB OPERANDS
                                        ;**********************************

001224  012701  001334          TEST5:  MOV     #T5DATA,R1
001230  005767  000100                  TST     T5DATA
001234  100020                          BPL     MSTART                          ;ERROR
001236  105767  000072                  TSTB    T5DATA
001242  001015                          BNE     MSTART                          ;ERROR
001244  105721                          TSTB    (R1)+
001246  001013                          BNE     MSTART                          ;ERROR
001250  105711                          TSTB    (R1)
001252  100011                          BPL     MSTART                          ;ERROR
001254  105741                          TSTB    -(R1)
001256  001007                          BNE     MSTART                          ;ERROR
001260  005721                          TST     (R1)+
001262  100005                          BPL     MSTART                          ;ERROR
001264  005711                          TST     (R1)
001266  001003                          BNE     MSTART                          ;ERROR
001270  005741                          TST     -(R1)
001272  100001                          BPL     MSTART                          ;ERROR
001274  000401                          BR      TEST6



001276  000000          MSTART: HALT                                            ;PROGRAM HALTS HERE
```

```
                                        ;*********************************
                                        ;* TEST6:
                                        ;* CHECK ALL BYTE DOUBLE OPERAND
                                        ;* INSTRUCTIONS/MODE ZERO
                                        ;*********************************

001300  012701  001340          TEST6:  MOV     #T6DATA,R1
001304  112102                          MOVB    (R1)+,R2
001306  141102                          BICB    (R1),R2
001310  131102                          BITB    (R1),R2
001312  001371                          BNE     MSTART                  ;ERROR
001314  151102                          BISB    (R1),R2
001316  124102                          CMPB    -(R1),R2
001320  001366                          BNE     MSTART                  ;ERROR
001322  000407                          BR      TEST7

001324  001324          T3DATA: .WORD   T3DATA
001326  001324                  .WORD   T3DATA
001330  177777          T3DAT4: .WORD   177777
001332  001330                  .WORD   T3DAT4
001334  100000          T5DATA: .WORD   100000
001336  000000                  .WORD   0
001340  125377          T6DATA: .WORD   125377
```

```
                                        ;**********************************
                                        ;* TEST7:
                                        ;* CHECK WORD INSTRUCTIONS WITH
                                        ;* MODE ZERO
                                        ;**********************************

001342  012702  001474          TEST7:  MOV     #T7DATA,R2
001346  016203  000002                  MOV     2(R2),R3
001352  012213                          MOV     (R2)+,(R3)
001354  024223                          CMP     -(R2),(R3)+
001356  001347                          BNE     MSTART                          ;ERROR

001360  005063  177776                  CLR     -2(R3)
001364  012232                          MOV     (R2)+,@(R2)+
001366  062243                          ADD     (R2)+,-(R3)
001370  164213                          SUB     -(R2),(R3)
001372  005272  177776                  INC     @-2(R2)
001376  005352                          DEC     @-(R2)
001400  044223                          BIC     -(R2),(R3)+
001402  001335                          BNE     MSTART                          ;ERROR

001404  012223                          MOV     (R2)+,(R3)+
001406  005443                          NEG     -(R3)
001410  031362  000002                  BIT     (R3),2(R2)
001414  001730                          BEQ     MSTART                          ;ERROR

001416  006213                          ASR     (R3)
001420  116213  000002                  MOVB    2(R2),(R3)
001424  005623                          SBC     (R3)+
001426  006163  177776                  ROL     -2(R3)
001432  001321                          BNE     MSTART                          ;ERROR

001434  056232  000002                  BIS     2(R2),@(R2)+
001440  006352                          ASL     @-(R2)
001442  006032                          ROR     @(R2)+
001444  000261                          SEC
001446  005552                          ADC     @-(R2)
001450  005332                          DEC     @(R2)+
001452  021252                          CMP     (R2),@-(R2)
001454  001310                          BNE     MSTART                          ;ERROR

001456  016243  000002                  MOV     2(R2),-(R3)
001462  005013                          CLR     (R3)
001464  005113                          COM     (R3)
001466  021362  177776                  CMP     (R3),-2(R2)
001472  001403                          BEQ     TEST8

001474  177777          T7DATA: .WORD   177777
001476  000500                  .WORD   000500
001500  000001                  .WORD   000001
```

```
                                        ;**********************************
                                        ;* TEST8:
                                        ;* CHECK ALL BYTE INSTRUCTIONS
                                        ;* MODES(DMBX)
                                        ;**********************************
001502  011203                  TEST8:  MOV     (R2),R3
001504  016213  177776                  MOV     -2(R2),(R3)             ;177777
001510  105013                          CLRB    (R3)                    ;177400 (DMB1) 0100
001512  105123                          COMB    (R3)+                   ;177777 (DMB2) 0000
001514  105632                          SBCB    @(R2)+                  ;177776 (DMB3) 1001
001516  105243                          INCB    -(R3)                   ;177777 (DMB4) 1000
001520  105452                          NEGB    @-(R2)                  ;177401 (DMB5) 0001
001522  105323                          DECB    (R3)+                   ;177400 (DMB2) 0101
001524  000363  177777                  SWAB    -1(R3)                  ;000377 (DMB6) 1000
001530  106332                          ASLB    @(R2)+                  ;000376 (DMB3) 1010
001532  106072  177776                  RORB    @-2(R2)                 ;000377 (DMB7) 1010
001536  106243                          ASRB    -(R3)                   ;000377 (DMB4) 1001
001540  106113                          ROLB    (R3)                    ;000377 (DMB1) 1001
001542  105523                          ADCB    (R3)+                   ;000000 (DMB2) 0101
001544  121343                          CMPB    (R3),-(R3)              ;000000 (DMB4) 0101
001546  001253                          BNE     MSTART                                  ;ERROR

001550  156252  177774                  BISB    -4(R2),@-(R2)           ;000377 (DMB5) 1001
001554  146223  177776                  BICB    -2(R2),(R3)+            ;000000 (DMB2) 0101
001560  136243  177776                  BITB    -2(R2),-(R3)            ;000000 (DMB4) 0101
001564  001401                          BEQ     LOOP
001566  000643                          BR      MSTART                                  ;ERROR


001570  000167  177254          LOOP:   JMP     TEST1

        001000                          .END CLEAR
```

```
LSI-11 CPU-DIAGNOSTICS MACRO V03.01 27-MAR-79 18:04:33 PAGE 6-1
SYMBOL TABLE

CLEAR   001000          LOOP    001570          TEST3   001142          TEST7   001342          T3DAT4  001330
CLR$    001020          MSTART  001276          TEST4   001200          TEST8   001502          T5DATA  001334
JMP1    001210          TEST1   001050          TEST5   001224          TIME    001046          T6DATA  001340
JMP3    001216          TEST2   001102          TEST6   001300          T3DATA  001324          T7DATA  001474

. ABS.  001574      000
        000000      001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:LSICPU=DK:LSICPU
```

3.1.2 LSI-11 MEMORY TEST

Source File: LSIMEM.MAC

Task File: LSIMEM.SAV

PURPOSE: To test LSI-11 Mos Memory board

DESCRIPTON: This program writes and reads various patterns into the LSI-11 memory starting at the last address of the program, to address 160000_8. The patterns used are all zeros, all ones, alternating ones and zeros and random numbers. If the data read from a memory location doesn't match the data written to it, the program halts displaying the current program address which is used to determine which test failed and location MSTART contains the bad memory address.

```
                                        .TITLE  MEMORY-DIAGNOSTICS
                                        .IDENT/V1.0/
000000                                  .ASECT
        001000                          .       =1000


                                        ;************************************************
                                        ;*  TESTS LSI-11 MEMORY (32KW)                  *
                                        ;*      1-WRITES AND READS ADDRESS INTO MEM.    *
                                        ;*      2-WRITES & READS ALL ZEROS              *
                                        ;*      3-WRITES & READS ALL ONES               *
                                        ;*      4-WRITES & READS ALT. ONES & ZEROS      *
                                        ;*      5-WRITES & READS ALT. ZEROS & ONES      *
                                        ;*      6-WRITES & READS RANDOM NUMBERS         *
                                        ;*  NOTE: PROGRAM HALTS ON A DETECTED ERROR     *
                                        ;*        SEE LISTING FOR CAUSE(PC), WITH       *
                                        ;*        (MSTART) HOLDING MEMORY ERROR         *
                                        ;*        LOCATION.                             *
                                        ;************************************************


001000  000005          CLEAR:  RESET                           ;RESET BUS
001002  012700  000340          MOV     #340,   R0              ;PRI=7
001006  106400                  MTPS    R0                      ;SET PS
001010  012706  000500          MOV     #500,   SP              ;INITIALIZE STACK
001014  012702  000400          MOV     #400,   R2              ;R2=TOP OF VECTOR AREA
001020  005042          CLR$:   CLR     -(R2)                   ;INSERT A HALT
Z 001022  010242                MOV     R2,     -(R2)           ;POINT VECTOR TO HALT
001024  005702                  TST     R2                      ;?R2=0 (DONE)
001026  001374                  BNE     CLR$                    ;NO CONTINUE
001030  012737  001000  000040  MOV     #CLEAR, @#40            ;RESTART ADDRESS
001036  012737  001046  000100  MOV     #TIME,  @#100           ;CLOCK RETURN
001044  000401                  BR      START                   ;BEGIN
001046  000002          TIME:   RTI                             ;IF CLOCK ON RETURN


                                        ;SEGMENT WRITES THE ADDRESS OF THE MEMORY ADDRESS
                                        ;INTO THAT ADDRESS

001050  013702  001426  START:  MOV     @#LOWLIM, R2            ;LOW ADDRESS OF MEMORY
001054  013703  001430          MOV     @#TOPLIM, R3            ;HIGHEST ADDRESS
Z 001060  010222          ADDRT:  MOV     R2,(R2)+                ;ADDRESS=ADDRESS
001062  020203                  CMP     R2,R3                   ;LOW=TOP
001064  101775                  BLOS    ADDRT
001066  024202          CHECK:  CMP     -(R2),R2                ;? ADDRESS=ADDRESS
001070  001403                  BEQ     ADCONT                  ;YES
001072  010237  001440          MOV     R2,     @#MSTART        ;NO ADDRESS AND HALT
001076  000000                  HALT                            ;***ERROR IN TEST
001100  020203          ADCONT: CMP     R2,     R3              ;TOP=LOW
001102  001371                  BNE     CHECK
```

```
                                        ;SEGMENT SELECTS MASKS AND CALLS WRITER
                                        ;AND READER
                                        
001104  012737  000000  001424  MASK1:  MOV     #0,     @#MASK                  ;MASK=000...0
001112  004767  000044                  JSR     PC,     RDWTM
001116  012737  177777  001424  MASK2:  MOV     #177777, @#MASK                 ;MASK=111...1
001124  004767  000032                  JSR     PC,     RDWTM
001130  012737  125252  001424  MASK3:  MOV     #125252, @#MASK                 ;MASK=1010...10
001136  004767  000020                  JSR     PC,     RDWTM
001142  012737  052525  001424  MASK4:  MOV     #52525,  @#MASK                 ;MASK=0101...01
001150  004767  000006                  JSR     PC,     RDWTM
001154  004767  000044          MASK5:  JSR     PC,RANDOM                       ;RANDOM NUMBERS

001160  000733                  BR      START                                   ;LOOP AROUND


                                        ;SEGMENT WRITES AND THEN READS BACK MASK
                                        ;HALTS ON ERROR

001162  013701  001424          RDWTM:  MOV     @#MASK, R1
001166  013702  001426                  MOV     @#LOWLIM,       R2
001172  013703  001430                  MOV     @#TOPLIM,       R3

001176  010122                  WRITE:  MOV     R1,     (R2)+
001200  020203                          CMP     R2,     R3                      ;LOW=TOP
001202  101775                          BLOS    WRITE
001204  024201                  READ:   CMP     -(R2),R1                        ;ADDRESS=MASK
001206  001403                          BEQ     MEMOK
001210  010237  001440                  MOV     R2,@#MSTART                     ;ERROR ADDRESS
001214  000000                          HALT                                    ;***ERROR IN TEST

001216  020203                  MEMOK:  CMP     R2,R3                           ;TOP=LOW
001220  001371                          BNE     READ
001222  000207                          RTS     PC
```

```
 1                                          ;SEGMENT GENERATES RANDOM NUMBERS AND WRITES THEN INTO
 2                                          ;MEMORY RESET SEEDS AND READS AND COMPARES MEMORY
 3
 4 001224  013703  001426          RANDOM: MOV     @#LOWLIM, R3                    ;LOW ADDRESS OF MEMORY
 5 001230  013704  001430                  MOV     @#TOPLIM,  R4                   ;HIGH LIMIT OF MEMORY
 6 001234  012737  176543  001432          MOV     #176543, @#HINUM                ;HI SEED
 7 001242  012737  123456  001434          MOV     #123456, @#LONUM                ;LOW SEED
 8
 9 001250  004767  000056          WRTRAN: JSR     PC,     GENRAN                  ;GENERATE RANDOM NUMBER
10 001254  013723  001436                  MOV     @#RANUM, (R3)+                  ;LOAD NUMBER
11 001260  020304                          CMP     R3,     R4                      ;?ALL OF MEMORY YET
12 001262  101772                          BLOS    WRTRAN
13 001264  012737  176543  001432  RDRAN:  MOV     #176543, @#HINUM                ;RESET SEED
14 001272  012737  123456  001434          MOV     #123456, @#LONUM                ;RESET SEED
15 001300  013703  001426                  MOV     @#LOWLIM,R3                     ;RESET LIMIT
16 001304  004767  000022          RD1:    JSR     PC,     GENRAN                  ;GENERATE RANDOM NUMBER
17 001310  022337  001436                  CMP     (R3)+,  @#RANUM                 ;MEM=RAN#
18 001314  001403                          BEQ     CONT
19 001316  010337  001440                  MOV     R3,     @#MSTART                ;ERROR IN MEMORY
20 001322  000000                          HALT                                    ;STOP CPU
21
22 001324  020304                  CONT:   CMP     R3,     R4                      ;?ALL OF MEMORY YET
23 001326  001366                          BNE     RD1
24 001330  000207                          RTS     PC
25
26                                         ;***** RANDOM NUMBER GENERATOR*******************
27
28 001332  010046                  GENRAN: MOV     R0,     -(SP)                   ;SAVE R0
29 001334  010146                          MOV     R1,     -(SP)                   ;SAVE R1
30 001336  010246                          MOV     R2,     -(SP)                   ;SAVE R2
31 001340  013700  001434                  MOV     @#LONUM,R0                      ;LOW VALUE
32 001344  013701  001432                  MOV     @#HINUM,R1                      ;HIGH VALUE
33 001350  012702  177771                  MOV     #-7,R2                          ;SHIFT COUNT
34 001354  006300                  G1$:    ASL     R0                              ;SHIFT R0 LEFT AND
35 001356  006101                          ROL     R1                              ;ROTATE CARRY INTO R1 AND
36 001360  005202                          INC     R2                              ;CHECK IF DONE
37 001362  001374                          BNE     G1$                             ;CONTINUE SHIFT LOOP
38 001364  063700  001434                  ADD     @#LONUM,R0                      ;ADD # TO MAKE X 129
39 001370  005501                          ADC     R1                              ;PROPOGATE CARRY
40 001372  062701  047401                  ADD     #47401,R1                       ;ADD HIGH CONSTANT
41 001376  010037  001434                  MOV     R0,     @#LONUM                 ;SAVE NUMBER
42 001402  010137  001432                  MOV     R1,     @#HINUM                 ;
43 001406  013737  001432  001436          MOV     @#HINUM,@#RANUM                 ;DEPOSIT RANDOM #
44 001414  012602                          MOV     (SP)+,R2                        ;POP STACK INTO R2
45 001416  012601                          MOV     (SP)+,R1                        ;POP STACK INTO R1
46 001420  012600                          MOV     (SP)+,R0                        ;POP STACK INTO R0
47 001422  000207                          RTS     PC
48
49                                         ;***********************************************
50
51 001424                          MASK:   .BLKW
52 001426  001442                  LOWLIM: .WORD   MSTART+2
53 001430  150000                  TOPLIM: .WORD   150000
54 001432  176543                  HINUM:  .WORD   176543
55 001434  123456                  LONUM:  .WORD   123456
56 001436                          RANUM:  .BLKW
57 001440                          MSTART: .BLKW
```

```
     58
     59        001000                          .END CLEAR
```

SYMBOL TABLE

ADCONT	001100		GENRAN	001332		MASK1	001104		MSTART	001440		READ	001204
ADDRT	001060		G1$	001354		MASK2	001116		RANDOM	001224		START	001050
CHECK	001066		HINUM	001432		MASK3	001130		RANUM	001436		TIME	001046
CLEAR	001000		LONUM	001434		MASK4	001142		RDRAN	001264		TOPLIM	001430
CLR$	001020		LOWLIM	001426		MASK5	001154		RLWTM	001162		WRITE	001176
CONT	001324		MASK	001424		MEMOK	001216		RD1	001304		WRTRAN	001250

. ABS. 001442 000
000002 001
ERRORS DETECTED: 2

VIRTUAL MEMORY USED: 288 WORDS (2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 51 PAGES
,DK:LSIMEM=DK:LSIMEM

3.1.3 BLIUI INTERFACE TEST

Source File: LIUINT.MAC

Task File: LIUINT.SAV

PURPOSE: To test the LSI-11/LIU interface

DESCRIPTION: This program tests the interface registers with patterns of all zeros and ones and continues by selecting INPUT BUFFER 0, where it performs a DMA write of pattern of alternating zeros and ones and then performs a DMA READ, testing the data read back. Should the data written not compare to the data read, the program halts, displaying the current program address.

```
                                        .TITLE  BLIUI INTERFACE DIAG.
                                        .IDENT  /V1.0/
                                        .ASECT
001000                                  .=      1000
172410                                  BAR=    172410
172412                                  WCR=    172412
172414                                  CSR=    172414
172416                                  OPREG=  172416

                                        ;*********************************************
                                        ;* DIAGNOSTIC FOR BUS LOOP INTERFACE UNIT    *
                                        ;* INTERFACE (BLIUI).                        *
                                        ;* (1)-CLEAR AND SET UP VECTORS              *
                                        ;* (2)-RESET INBUF0 POINTER                  *
                                        ;* (3)-DMA 256 BYTES TO LIU                  *
                                        ;*     DMA 256 BYTES FROM LIU                *
                                        ;* (4)-COMPARE REC. DATA TO SENT DATA HALT   *
                                        ;*      IF NOT THE SAME.                     *
                                        ;*********************************************

                                        ;CLEAR INTERRUPT AND TRAP VECTOR AREA

001000  000005                 CLEAR:  RESET                           ;RESET BUS
001002  012700  000340                  MOV     #340,   R0              ;PRI=7
001006  106400                          MTPS    R0                      ;SET PS
001010  012706  000500                  MOV     #500,   SP              ;INITIALIZE STACK
001014  012702  000200                  MOV     #200,   R2              ;R2=TOP OF VECTOR AREA
001020  005042                 CLR$:   CLR     -(R2)                   ;INSERT HALT AT VECTOR
001022  010242                          MOV     R2,     -(R2)           ;POINT TO HALT
001024  005702                          TST     R2                      ;R2=0 (DONE)
001026  001374                          BNE     CLR$                    ;NO CONTINUE
001030  012737  001000  000040          MOV     #CLEAR, @#40            ;RESTART ADDRESS
001036  012737  001064  000100          MOV     #TIME,  @#100           ;CLOCK
001044  052737  000001  172414          BIS     #1,     @#CSR           ;INIT LIU
001052  000240                          NOP                             ;DELAY
001054  042737  000001  172414          BIC     #1,     @#CSR           ;CLEAR INIT
001062  000401                          BR      RUN                     ;BEGIN
001064  000002                 TIME:   RTI                             ;IF CLOCK ON RETURN

                                        ;SELECT MASK TO WRITE AND READ BACK
                                        ; CALL FOR POINTER RESET
                                        ; SET UP DATA BLOCK

001066  005037  001756          RUN:    CLR     @#FLAG                  ;CLEAR LOCAL STATUS
001072  042737  040100  172414          BIC     #040100,@#CSR           ;DISABLE INTERRUPTS
001100  004767  000502                  JSR     PC,     BUFPNT          ;ZERO BUFFER POINTER
001104  022737  000001  001756  TST:    CMP     #1,     @#FLAG          ;TST1 COMPLETE?
001112  001417                          BEQ     TST2                    ;YES
001114  012700  001762          TST1:   MOV     #SIDATA, R0             ;DATA BLOCK ADDRESS
001120  012737  125125  001760          MOV     #125125, @#MASK         ;10101010..10
001126  012737  000001  001756          MOV     #1,     @#FLAG          ;SET LOCAL STATUS
001134  012702  177600                  MOV     #-128., R2              ;# OF DATA BYTES
001140  013720  001760          T1$:    MOV     @#MASK, (R0)+           ;DEPOSIT DATA
001144  005202                          INC     R2                      ;256 YET
001146  001374                          BNE     T1$                     ;NO-CONTINUE
001150  000416                          BR      REC
```

```
  1 001152  012700  001762          TST2:   MOV     #SDATA, R0              ;DATA BLOCK ADDRESS
  2 001156  012737  052652  001760          MOV     #052652, @#MASK         ;010101..01
  3 001164  005037  001756                   CLR     @#FLAG                  ;CLEAR LOCAL STATUS
  4 001170  012702  177600                   MOV     #-128., R2              ;# FO DATA BYTES
  5 001174  013720  001760          T2$:    MOV     @#MASK, (R0)+           ;DEPOSIT DATA
  6 001200  005202                           INC     R2                      ;256 TIMES
  7 001202  001374                           BNE     T2$
  8 001204  000400                           BR      REG
  9
 10                                         ;READ AND WRITE REGISTERS
 11
 12 001206  012737  177777  172410  REG:    MOV     #177777, @#BAR          ;WRITE ALL ONES
 13 001214  013700  172410                   MOV     @#BAR, R0               ;READ
 14 001220  022700  177777                   CMP     #177777, R0             ;WAS IT THE SAME
 15 001224  001401                           BEQ     1$                      ;YES
 16 001226  000000                           HALT
 17 001230  012737  000000  172410  1$:     MOV     #000000, @#BAR          ;WRITE ALL ZEROS
 18 001236  013700  172410                   MOV     @#BAR, R0               ;READ
 19 001242  022737  000000  172410           CMP     #0, @#BAR               ;SAME
 20 001250  001401                           BEQ     2$                      ;YES
 21 001252  000000                           HALT                            ;NO ERROR
 22 001254  012737  177777  172412  2$:     MOV     #177777, @#WCR          ;WRITE ALL ONES WCR
 23 001262  013700  172412                   MOV     @#WCR , R0              ;READ
 24 001266  022700  177777                   CMP     #177777, R0             ;SAME
 25 001272  001401                           BEQ     3$                      ;YES
 26 001274  000000                           HALT                            ;NO ERROR
 27 001276  012737  000000  172412  3$:     MOV     #000000, @#WCR          ;WRITE ALL ONE WCR
 28 001304  013700  172412                   MOV     @#WCR, R0               ;READ
 29 001310  022700  000000                   CMP     #0, R0                  ;SAME
 30 001314  001401                           BEQ     4$                      ;YES
 31 001316  000000                           HALT                            ;NO HALT
 32 001320  052737  040100  172414  4$:     BIS     #040100, @#CSR          ;SET R/W BITS
 33 001326  032737  040100  172414           BIT     #040100, @#CSR          ;BITS ON
 34 001334  001001                           BNE     5$                      ;YES
 35 001336  000000                           HALT                            ;ERROR
 36 001340  005037  172414          5$:     CLR     @#CSR                   ;CLAER BITS
 37 001344  032737  040100  172414           BIT     #040100, @#CSR          ;BITS ON
 38 001352  001401                           BEQ     SEND                    ;NO
 39 001354  000000                           HALT                            ;YES ERROR
 40
```

```
                                        ;SENDS 256 BYTES TO LIU BUFFERS
                                        ;RECEIVED 256 BYTES (DMA)
                                        
001356  012737  010440  172416  SEND:   MOV     #4384., @#OPREG         ;SEL BUFFER
001364  012737  001762  172410          MOV     #SIATA, @#BAR           ;DATA ADDRESS
001372  012705  000400                  MOV     #256.,  R5              ;BYTE COUNT
001376  005405                          NEG     R5                      ; 2'S COMP
001400  010537  172412                  MOV     R5,     @#WCR
001404  012737  024000  172416          MOV     #10240., @#OPREG        ;DMA GOES
001412  000240                          NOP                             ;DELAY NOP
001414  105737  172414          SDMA:   TSTB    @#CSR                   ;WAS IT A GOOD XFER
001420  100401                          BMI     RECV                    ;YES
001422  000000                  ES$:    HALT                            ;NO
                                        
001424  105037  172414          RECV:   CLRB    @#CSR
001430  012737  001400  172416          MOV     #766.,  @#OPREG         ;FALSE READ (POINTER)
001436  105737  172414                  TSTB    @#CSR                   ;DONE BIT ON?
001442  100375                          BPL     .-4
001444  105037  172414          RV1$:   CLRB    @#CSR
001450  012737  010440  172416          MOV     #4384., @#OPREG         ;SEL BUFFER
001456  012737  002364  172410          MOV     #RDATA, @#BAR           ;DATA BLOCK
001464  012705  000400                  MOV     #256.,  R5              ;COUNT
001470  005405                          NEG     R5                      ; 2'S COMP
001472  010537  172412                  MOV     R5,     @#WCR
001476  012737  021000  172416          MOV     #8704., @#OPREG         ;DMA READ GONE
001504  000240                          NOP
001506  105737  172414          RDMA:   TSTB    @#CSR                   ;WAS IT A GOOD XFER
001512  100375                          BPL     .-4
                                        
                                        ;SWAP DATA BYTES THAT WERE RECEIVED
                                        
001514  105037  172414          SWAP:   CLRB    @#CSR
001520  012700  002364                  MOV     #RDATA, R0              ;ADDRESS OF DATA
001524  012701  177600                  MOV     #-128., R1              ;WORD COUNT
001530  000320                  SWB:    SWAB    (R0)+                   ;SWAP BYTES
001532  005201                          INC     R1
001534  001375                          BNE     SWB                     ;128 TIMES
                                        
                                        ;TEST DATA THAT HAS BEEN RECIEVED
                                        
001536  013700  001760          TSTDAT: MOV     @#MASK, R0              ;MASK IN USE
001542  012701  002364                  MOV     #RDATA, R1              ;DATA BLOCK
001546  012702  177600                  MOV     #-128., R2              ;# OF BYTES
001552  020021                  CMP:    CMP     R0,(R1)+                ;MASK=DATA
001554  001401                          BEQ     INCT
001556  000000                          HALT                            ;ERROR IN DATA
001560  005202                  INCT:   INC     R2
001562  001373                          BNE     CMP
                                        
001564  012700  177600          CLRRD:  MOV     #-128.,R0               ;CLEAR RECIEVED DATA
001570  012701  002364                  MOV     #RDATA, R1
001574  005021                  C1$:    CLR     (R1)+
001576  005200                          INC     R0
001600  001375                          BNE     C1$
001602  000167  177276                  JMP     TST
```

```
                                        ;CLEAR INPUT BUFFER Ø POINTER
                                        
001606  012700  010410          BUFPNT: MOV     #4360., RØ              ;RDBUFPT COMMAND
001612  012701  010440                  MOV     #4384., R1              ;INBUFØ
001616  010037  172416                  MOV     RØ,     @#OPREG         ;BUFFER POINTER
001622  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD
001630  105737  172414                  TSTB    @#CSR                   ;'DONE' BIT ON?
001634  100375                          BPL     .-4
001636  105037  172414          B1$:    CLRB    @#CSR
001642  013702  172416                  MOV     @#OPREG, R2             ;FETCH DATA
001646  042702  177400                  BIC     #177400, R2             ;CLEAR UNUSED BITS
001652  010137  172416                  MOV     R1,     @#OPREG         ;SEL BUFFER
001656  022702  000000          CP1$:   CMP     #Ø,     R2              ;POINTER?=Ø
001662  001412                          BEQ     CKBP                    ;YES
001664  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD
001672  105737  172414                  TSTB    @#CSR
001676  100375                          BPL     .-4
001700  105037  172414          B2$:    CLRB    @#CSR
001704  005302                          DEC     R2                      ;UNTIL ZERO
001706  000763                          BR      CP1$
001710  010037  172416          CKBP:   MOV     RØ,     @#OPREG         ;RE-READ POINTER
001714  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD
001722  105737  172414                  TSTB    @#CSR                   ;'DONE' BIT ON?
001726  100375                          BPL     .-4
001730  105037  172414          B3$:    CLRB    @#CSR
001734  013702  172416                  MOV     @#OPREG, R2             ;FETCH DATA
001740  042702  177400                  BIC     #177400, R2             ;CLEAR UNUSED BITS
001744  022702  000000                  CMP     #Ø,     R2              ;POINTER?=Ø
001750  001001                          BNE     EBPT
001752  000207                          RTS     PC                      ;GOOD RETURN
001754  000000          EBPT:   HALT                                    ;POINTER DOES NOT=Ø
                                        
                                        ;DATA BLOCK
                                        
001756  000000          FLAG:   .WORD Ø
001760                  MASK:   .BLKW
001762                  SDATA:  .BLKB   256.
002362  177777          EOBS:   .WORD   177777
002364                  RDATA:  .BLKB   256.
002764  177777          EOBR:   .WORD   177777
                                .EVEN
                                
001000                          .END CLEAR
```

```
BLIUI INTERFACE DIAG.  MACRO V03.01 27-MAR-79 18:03:04 PAGE 4-1
SYMBOL TABLE

BAR    = 172410          CLR$    001020          ES$     001422          REG     001206          TIME    001064
BUFPNT  001606           CMP     001552          FLAG    001756          RUN     001066          TST     001104
B1$     001636           CP1$    001656          INCT    001560          RV1$    001444          TSTDAT  001536
B2$     001700           CSR    = 172414         MASK    001760          SDATA   001762          TST1    001114
B3$     001730           C1$     001574          OPREG  = 172416          SDMA    001414          TST2    001152
CKBP    001710           EBPT    001754          RDATA   002364          SEND    001356          T1$     001140
CLEAR   001000           EOBR    002764          RDMA    001506          SWAP    001514          T2$     001174
CLRRD   001564           EOBS    002362          RECV    001424          SWB     001530          WCR    = 172412

. ABS.  002766     000
        000000     001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:LIUINT=DK:LIUINT
```

3.1.4 LIU ADDRESS COMPARISON RAM TEST

Source File: LIURAM.MAC

Task File: LIURAM.SAV

PURPOSE: To test the address comparison RAM of the LIU

DESCRIPTION: This program writes and reads various patterns to the address comparison RAM. The patterns used are all zeros, all ones, alternating ones and zeros and random numbers. If the data read from a memory location doesn't match the data written to it, the program halts displaying current program address. Otherwise, the program loops, repeating the test indefinitely.

```
                                        .TITLE  ACRAM-DIAGNOSTICS
                                        .IDENT  /V1.0/
000000                                  .ASECT
        001000                          =1000
        172416                          OPREG=  172416
        172414                          CSR=    172414


                                        ;************************************************
                                        ;*  TESTS LIU ACRAM MEMORY                      *
                                        ;*      TEST1- WRITE & READ 0000                *
                                        ;*      TEST2- WRITE & READ 1111                *
                                        ;*      TEST3- WRITE & READ 1010                *
                                        ;*      TEST4- WRITE & READ 0101                *
                                        ;*      TEST5- RANDOM NUMBERS                   *
                                        ;*      RESET- CHECK POINTER                    *
                                        ;*  NOTE: PROGRAM HALTS ON A DETECTED ERROR     *
                                        ;*        SEE LISTING FOR CAUSE(PC).            *
                                        ;************************************************


001000  000005                  CLEAR:  RESET                           ;RESET BUS
001002  012701  000340                  MOV     #340,   R1              ;PRI=7
001006  106400                          MTPS    R0                      ;SET PS
001010  012706  000500                  MOV     #500,   SP              ;INITIALIZE STACK
001014  012702  000400                  MOV     #400,   R2              ;R2=TOP OF VECTOR AREA
001020  005042                  CLR$:   CLR     -(R2)                   ;INSERT A HALT
Z 001022  010242                        MOV     R2,     -(R2)           ;POINT VECTOR TO HALT
001024  005702                          TST     R2                      ;?R2=0 (DONE)
001026  001374                          BNE     CLR$                    ;NO CONTINUE
001030  012737  001000  000040          MOV     #CLEAR, @#40            ;RESTART ADDRESS
001036  012737  000100  000100          MOV     #100,   @#100           ;CLOCK
001044  052737  000001  172414          BIS     #1,     @#CSR           ;INIT LIU
001052  000240                          NOP
001054  042737  000001  172414          BIC     #1,     @#CSR           ;RESET LIU
001062  000401                          BR      TEST1                   ;BEGIN
001064  000002                  TIME:   RTI                             ;IF CLOCK ON RETURN

                                        ;WRITE ALL ZEROS

001066  004567  001024          TEST1:  JSR     R5,RESET
001072  012700  000400                  MOV     #256.,  R0              ;COUNTER
001076  012737  010401  172416          MOV     #4353., @#OPREG         ;ACRAM SELECT
001104  012737  004400  172416  W1$:    MOV     #4400,  @#OPREG         ;WD
001112  105737  172414                  TSTB    @#CSR                   ;VALID RD
001116  100375                          BPL     .-4                     ;NO LOOP UNTIL
001120  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
001124  077011                          SOB     R0,     W1$             ;256 TIMES
```

```
                                        ;READ ALL ZEROS
                                        
001126  004567  000764                  JSR     R5,     RESET                   ;RESET ADDRESS
001132  005000                          CLR     R0
001134  012737  010401  172416          MOV     #4353., @#OPREG                 ;ACRAM SELECT
001142  012737  001400  172416  R1$:    MOV     #768.,  @#OPREG                 ;RD
001150  105737  172414                  TSTB    @#CSR                           ;VALID WD
001154  100375                          BPL     .-4                             ;NO LOOP UNTIL
001156  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001162  013703  172416                  MOV     @#OPREG, R3                     ;FETCH DATA
001166  042703  177760                  BIC     #177760, R3                     ;CLEAR UNUSED BITS
001172  022703  000000                  CMP     #0,      R3                     ;DATA=0
001176  001005                          BNE     E1$                             ;NO
001200  005200                          INC     R0
001202  022700  000400                  CMP     #256.,  R0                      ;ALL YET (256)
001206  001355                          BNE     R1$
001210  000401                          BR      TEST2
001212  000000                  E1$:    HALT                                    ;ERROR IN DATA

                                        ;WRITE ALL ONES

001214  004567  000676          TEST2:  JSR     R5,     RESET                   ;RESET ADDRESS
001220  012700  000400                  MOV     #256.,  R0                      ;COUNTER
001224  012737  010401  172416          MOV     #4353., @#OPREG                 ;ACRAM SELECT
001232  012737  004417  172416  W2$:    MOV     #4417,  @#OPREG                 ;WD MASK=ALL ZEROS
001240  105737  172414                  TSTB    @#CSR                           ;VALID WD
001244  100375                          BPL     .-4                             ;NO LOOP UNTIL
001246  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001252  077011                          SOB     R0,     W2$

                                        ;READ ALL ONES
001254  004567  000636                  JSR     R5,     RESET                   ;ADDRESS=0
001260  005000                          CLR     R0
001262  012737  010401  172416          MOV     #4353., @#OPREG                 ;SELECT ACRAM
001270  012737  001400  172416  R2$:    MOV     #768.,  @#OPREG                 ;RD COMMAND
001276  105737  172414                  TSTB    @#CSR                           ;VALID RD
001302  100375                          BPL     .-4                             ;NO LOOP UNTIL
001304  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001310  013703  172416                  MOV     @#OPREG, R3                     ;FETCH DATA
001314  042703  177760                  BIC     #177760, R3                     ;CLEAR UNUSED BITS
001320  022703  000017                  CMP     #17,     R3                     ;DATA=1111
001324  001005                          BNE     E2$                             ;NO
001326  005200                          INC     R0
001330  022700  000400                  CMP     #256.,  R0
001334  001355                          BNE     R2$
001336  000401                          BR      TEST3
001340  000000                  E2$:    HALT                                    ;ERROR IN DATA

                                        ;WRITE 1010
001342  004567  000550          TEST3:  JSR     R5,     RESET                   ;ADDRESS=0
001346  016700  177026                  MOV     256.,   R0                      ;COUNTER
001352  012737  010401  172416          MOV     #4353., @#OPREG                 ;SELECT ACRAM
001360  012737  004412  172416  W3$:    MOV     #4412,  @#OPREG                 ;WD MASK=1010
001366  105737  172414                  TSTB    @#CSR                           ;VALID WD
001372  100375                          BPL     .-4                             ;NO LOOP UNTIL
001374  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001400  077011                          SOB     R0,     W3$                     ;LOOP 256
```

```
                                        ;READ   1010
                                        
001402  004567  000510                  JSR     R5,     RESET           ;ADDRESS=0
001406  005000                          CLR     R0
001410  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
001416  012737  001400  172416  R3$:    MOV     #768.,  @#OPREG         ;RD COMMAND
001424  105737  172414                  TSTB    @#CSR                   ;VALID RD
001430  100375                          BPL     .-4                     ;NO LOOP UNTIL
001432  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
001436  013703  172416                  MOV     @#OPREG, R3             ;FETCH DATA
001442  042703  177760                  BIC     #177760, R3             ;CLEAR UNUSED BITS
001446  022703  000012                  CMP     #12,    R3              ;DATA=1010
001452  001005                          BNE     E3$                     ;NO
001454  005200                          INC     R0
001456  022700  000400                  CMP     #256.,  R0              ;256 TIMES
001462  001355                          BNE     R3$
001464  000401                          BR      TEST4
001466  000000                  E3$:    HALT                            ;ERROR IN DATA
                                        
                                        ;WRITE  0101
                                        
001470  004567  000422          TEST4:  JSR     R5,     RESET           ;ADDRESS=0
001474  012700  000400                  MOV     #256.,  R0
001500  012737  010401  172416          MOV     #4353., @#OPREG         ;COMMAND SEL ACRAM
001506  012737  004405  172416  W4$:    MOV     #4405,  @#OPREG         ;WD MASK=0101
001514  105737  172414                  TSTB    @#CSR                   ;VALID WD
001520  100375                          BPL     .-4                     ;NO LOOP UNTIL
001522  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
001526  077011                          SOB     R0,     W4$             ;LOOP
                                        
                                        ;READ   0101
                                        
001530  004567  000362                  JSR     R5,     RESET           ;RESET POINTER
001534  005000                          CLR     R0
001536  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
001544  012737  001400  172416  R4$:    MOV     #768.,  @#OPREG         ;RD
001552  105737  172414                  TSTB    @#CSR                   ;VALID RD
001556  100375                          BPL     .-4                     ;NO LOOP UNTIL
001560  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
001564  013703  172416                  MOV     @#OPREG, R3             ;FETCH DATA
001570  042703  177760                  BIC     #177760, R3             ;CLEAR UNUSED BITS
001574  022703  000005                  CMP     #5,     R3              ;DATA=0101
001600  001005                          BNE     E4$                     ;NO
001602  005200                          INC     R0
001604  022700  000400                  CMP     #256.,  R0              ;256 TIMES
001610  001355                          BNE     R4$
001612  000401                          BR      TEST5
001614  000000                  E4$:    HALT                            ;ERROR IN DATA
```

```
 1                                          ;WRITE RANDOM NUMBERS
 2
 3 001616  004567  000274          TEST5:  JSR     R5,     RESET           ;RESET POINTER
 4 001622  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
 5 001630  012705  000400                  MOV     #256.,  R5              ;COUNTER
 6 001634  012737  176543  002110          MOV     #176543, @#HINUM        ;RANDOM SEED
 7 001642  012737  123456  002112          MOV     #123456, @#LONUM        ;RANDOM SEED
 8
 9 001650  004767  000142          W5$:    JSR     PC,     GENRAN          ;GENERATE RANDOM NUMBER
10 001654  013704  002114                  MOV     @#RANUM, R4             ;FETCH NUMBER
11 001660  062704  004400                  ADD     #4400,   R4             ;ADD WD COMMAND
12 001664  010437  172416                  MOV     R4,      @#OPREG        ;WRITE DATA ACRAM
13 001670  105737  172414                  TSTB    @#CSR                   ;VALID WD
14 001674  100375                          BPL     .-4                     ;NO LOOP UNTIL
15 001676  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
16 001702  077516                          SOB     R5,     W5$             ;LOOP 256X
17
18 001704  004567  000206                  JSR     R5,     RESET           ;RESET ACRAM POINTER
19 001710  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
20 001716  005005                          CLR     R5                      ;CLEAR LOOP COUNTER
21 001720  012737  176543  002110          MOV     #176543, @#HINUM        ;RESET SEED
22 001726  012737  123456  002112          MOV     #123456, @#LONUM        ;RESET SEED
23
24 001734  004767  000056          R5$:    JSR     PC,     GENRAN          ;GENERATE RANDOM NUMBER
25 001740  013704  002114                  MOV     @#RANUM, R4             ;FETCH NUMBER
26 001744  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD LIU
27 001752  105737  172414                  TSTB    @#CSR                   ;VALID RD
28 001756  100375                          BPL     .-4                     ;NO LOOP UNTIL
29 001760  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
30 001764  013703  172416                  MOV     @#OPREG, R3             ;FETCH DATA
31 001770  042703  177760                  BIC     #177760, R3             ;CLEAR UNUSED BITS
32 001774  020304                          CMP     R3,     R4              ;DATA=RAN NUMBER GEN'ED
33 001776  001006                          BNE     E5$
34 002000  005205                          INC     R5                      ;INCREMENT POINTER
35 002002  022705  000400                  CMP     #256.,  R5              ;DONE
36 002006  001352                          BNE     R5$                     ;NO
37 002010  000167  177052                  JMP     TEST1                   ;YES START AGAIN
38 002014  000000                  E5$:    HALT                            ;ERROR IN DATA
39
```

```
                                        ;*******RANDOM NUMBER GENERATOR
                                        
002016  013700  002112          GENRAN: MOV     @#LONUM, R0             ;SEED 1
002022  013701  002110                  MOV     @#HINUM, R1             ;SEED 2
002026  012702  177771                  MOV     #-7,    R2              ;SHIFT COUNT
002032  006300                  GR:     ASL     R0                      ;SHIFT R0 LEFT AND
002034  006101                          ROL     R1                      ;ROTATE CARRY INTO R1
002036  005202                          INC     R2                      ;INC AND CHECK FOR 0
002040  001374                          BNE     GR
002042  063700  002112                  ADD     @#LONUM, R0             ;ADD NUMBER TO MAKE  129
002046  005501                          ADC     R1                      ;PROPOGATE CARRY
002050  063701  002110                  ADD     @#HINUM, R1             ;ADD NUMBER TO 129
002054  062700  001057                  ADD     #1057,  R0              ;ADD LOW CONSTANT
002060  005501                          ADC     R1                      ;PROPOGATE CARRY
002062  062701  047401                  ADD     #47401, R1              ;ADD HIGH CONSTANT
002066  010037  002112                  MOV     R0,     @#LONUM         ;SAVE R0
002072  010137  002110                  MOV     R1,     @#HINUM         ;SAVE R1
002076  042701  177760                  BIC     #177760, R1             ;CLEAR HIGH BITS FOR AR
002102  010137  002114                  MOV     R1,     @#RANUM         ;NUMBER GENERATED
002106  000207                          RTS     PC                      ;RETURN

002110  176543                  HINUM:  .WORD   176543
002112  123456                  LONUM:  .WORD   123456
002114                          RANUM:  .BLKW

                                        ;********

                                        ;RESET ACRAM POINTER

002116  012737  010402  172416  RESET:  MOV     #4354., @#OPREG         ;LOAD ADDRESS
002124  012737  004400  172416          MOV     #2304., @#OPREG         ;WD 0
002132  105737  172414                  TSTB    @#CSR                   ;VALID WD
002136  100375                          BPL     .-4                     ;NO LOOP UNTIL
002140  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
002144  000205                          RTS     R5

002146                          MSTART: .BLKW

        001000                          .END CLEAR
```

```
ACRAM-DIAGNOSTICS       MACRO V03.01 27-MAR-79 18:02:25 PAGE 5-1
SYMBOL TABLE

CLEAR   001000          E5$     002014          RANUM   002114          R5$     001734          TIME    001064
CLR$    001020          GENRAN  002016          RESET   002116          TEST1   001066          W1$     001104
CSR   = 172414          GR      002032          R1$     001142          TEST2   001214          W2$     001232
E1$     001212          HINUM   002110          R2$     001270          TEST3   001342          W3$     001360
E2$     001340          LONUM   002112          R3$     001416          TEST4   001470          W4$     001506
E3$     001466          MSTART  002146          R4$     001544          TEST5   001616          W5$     001650
E4$     001614          OPREG = 172416

. ABS.  002150      000
        000000      001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:LIURAM=DK:LIURAM
```

3.1.5 LIU BUFFER TEST

Source File: LIUBUF.MAC

Task File: LIUBUF.SAV

PURPOSE: To test the input and output buffers in the LIU

DESCRIPTION: This program initializes the buffers and then writes and reads various patterns from the LIU buffers. The patterns used are all ones, all zeros, alternating ones and zeros and random numbers. If the data read from the LIU buffer does not match the data written to it, the program stores an indicator of which buffer is in error and halts, displaying the current program address. Otherwise, the program loops, repeating the test indefinitely.

```
                                        .TITLE  BUFFER-DIAGNOSTICS
                                        .IDENT  /V1.0/
000000                                  .ASECT
        001000                          =1000
        172416                          OPREG=  172416
        172414                          CSR=    172414


                                        ;************************************************
                                        ;*  TESTS LIU INPUT & OUTPUT BUFFERS            *
                                        ;*   CLEAR- TRAP INITIALIZE                     *
                                        ;*   START- SELECTS BUFFERS                     *
                                        ;*     RUN- SELECTS TESTS & MASKS               *
                                        ;*   TEST1- CHECKS BUFFER POINTERS              *
                                        ;*   TEST2- WRITES & READS MASKS                *
                                        ;*   TEST3- WRITES & READS RANDOM NUMBERS       *
                                        ;*  GENRAN- GENERATES RANDOM NUMBERS            *
                                        ;*  NOTE: PROGRAM HALTS ON A DETECTED ERROR     *
                                        ;*        SEE LISTING FOR CAUSE(PC).            *
                                        ;************************************************


001000  000005          CLEAR:  RESET                           ;RESET BUS
001002  012700  000340          MOV     #340,   R0              ;PRI=7
001006  106400                  MTPS    R0                      ;SET PS
001010  012706  000500          MOV     #500,   SP              ;INITIALIZE STACK
001014  012702  000400          MOV     #400,   R2              ;R2=TOP OF VECTOR AREA
001020  005042          CLR$:   CLR     -(R2)                   ;INSERT A HALT
Z   001022  010242              MOV     R2,     -(R2)           ;POINT VECTOR TO HALT
001024  005702                  TST     R2                      ;?R2=0 (DONE)
001026  001374                  BNE     CLR$                    ;NO CONTINUE
001030  012737  001000  000040  MOV     #CLEAR, @#40            ;RESTART ADDRESS
001036  012737  001064  000100  MOV     #TIME,  @#100           ;CLOCK
001044  052737  000001  172414  BIS     #1,     @#CSR           ;CLEAR LIU
001052  000240                  NOP
001054  042737  000001  172414  BIC     #1,     @#CSR           ;RESET LIU
001062  000401                  BR      START                   ;BEGIN
001064  000002          TIME:   RTI                             ;IF CLOCK ON RETURN


                                        ;SEGMENT SELECTS ONE OF THE FOUR BUFFER
                                        ;AND THEN CALLS SEGMENT RUN

001066  012737  010440  002022  START:  MOV     #4384., @#SELBUF        ;COMMAND INBUF0
001074  012737  010410  002024          MOV     #4360., @#BFPCMD        ;COMMAND RDBUFP
001102  004767  000062                  JSR     PC,     RUN
001106  012737  010640  002022          MOV     #4512., @#SELBUF        ;COMMAND INBUF1
001114  012737  010610  002024          MOV     #4488., @#BFPCMD        ;COMMAND RDBUFP
001122  004767  000042                  JSR     PC,     RUN
001126  012737  010540  002022          MOV     #4448., @#SELBUF        ;COMMAND OUTBUF0
001134  012737  010510  002024          MOV     #4424., @#BFPCMD        ;COMMAND RDBUFP
001142  004767  000022                  JSR     PC,     RUN
001146  012737  010740  002022          MOV     #4576., @#SELBUF        ;COMMAND OUTBUF1
001154  012737  010710  002024          MOV     #4552., @#BFPCMD        ;COMMAND RDBUFP
001162  004767  000002                  JSR     PC,     RUN
001166  000737                  BR      START                   ;LOOP AROUND
```

58

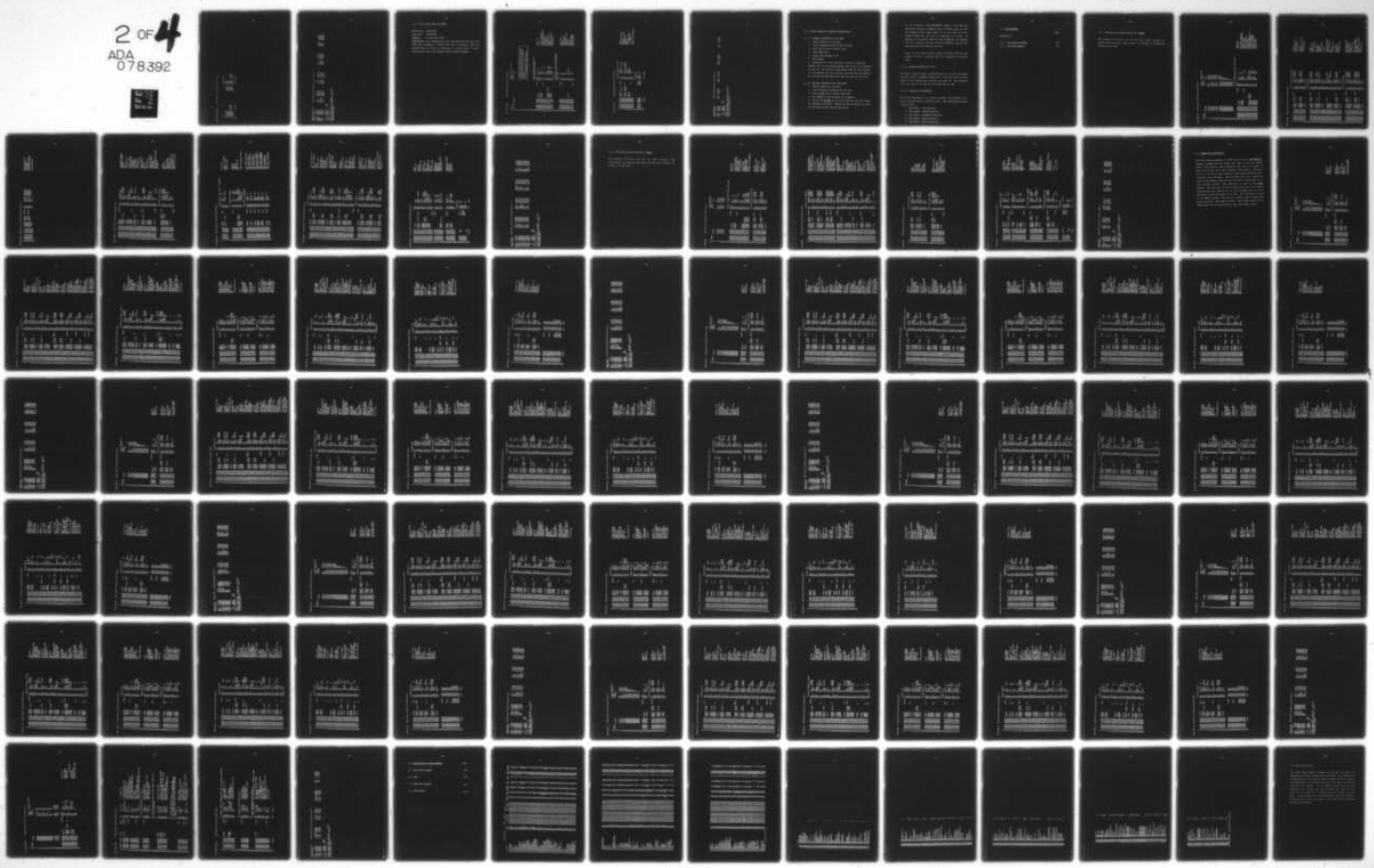
```
                                        ;SEGMENT CONTROLS WHICH TEST AND MASKS
                                        ;THAT ARE SENT TO THE READER/WRITERS
001170  004767  000056          RUN:  JSR     PC,     TEST1                   ;TEST BUFFER POINTERS
001174  012737  000000  002026        MOV     #0,     @#MASK                  ;MASK-00000000
001202  004767  000220                JSR     PC,     TEST2                   ;WRITE/READ MASK
001206  012737  000377  002026        MOV     #377,   @#MASK                  ;MASK-11111111
001214  004767  000206                JSR     PC,     TEST2                   ;WRITE/READ MASK
001220  012737  000252  002026        MOV     #252,   @#MASK                  ;MASK-10101010
001226  004767  000174                JSR     PC,     TEST2                   ;WRITE/READ MASK
001232  012737  000125  002026        MOV     #125,   @#MASK                  ;MASK-01010101
001240  004767  000162                JSR     PC,     TEST2                   ;WRITE/READ MASK
001244  004767  000306                JSR     PC,     TEST3                   ;RANDOM NUMBER TEST
001250  000207                       RTS     PC                              ;RETURN TO START


                                        ;TEST CHECKS TO SEE THAT BUFFER POINTERS CAN BE
                                        ;SET TO ZERO. HALT ON ERROR

001252  013700  002024          TEST1:  MOV     @#BFPCMD, R0                    ;COMMAND READ POINTER
001256  013701  002022                  MOV     @#SELBUF, R1                    ;SEL BUFFERS
001262  010037  172416                  MOV     R0,     @#OPREG                 ;RDBUFADR
001266  012737  001400  172416          MOV     #768.,  @#OPREG                 ;RD
001274  105737  172414                  TSTB    @#CSR                           ;VALID  RD
001300  100375                         BPL     .-4                             ;NO LOOP UNTIL
001302  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001306  013702  172416                  MOV     @#OPREG, R2                     ;FETCH DATA
001312  042702  177400                  BIC     #177400, R2                     ;CLEAR BITS
001316  013737  002022  172416          MOV     @#SELBUF, @#OPREG               ;SEL BUFFER
001324  022702  000000          C1$:    CMP     #0,     R2                      ;POINTER=0
001330  001412                         BEQ     CKBP
001332  012737  001400  172416          MOV     #768.,  @#OPREG                 ;RD
001340  105737  172414                  TSTB    @#CSR                           ;VALID RD
001344  100375                         BPL     .-4                             ;NO LOOP UNTIL
001346  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001352  005302                         DEC     R2                              ;DECREMENT COUNT
001354  000763                         BR      C1$
001356  013737  002024  172416  CKBP:  MOV     @#BFPCMD,@#OPREG                ;RDBUFADR
001364  012737  001400  172416          MOV     #768.,  @#OPREG                 ;RD
001372  105737  172414                  TSTB    @#CSR                           ;VALID RD
001376  100375                         BPL     .-4                             ;NO LOOP UNTIL
001400  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001404  013702  172416                  MOV     @#OPREG, R2                     ;FETCH DATA
001410  042702  177400                  BIC     #177400, R2                     ;CLEAR BITS
001414  022702  000000                  CMP     #0,      R2                     ;POINTER=0
001420  001001                         BNE     E1$                             ;NO
001422  000207                         RTS     PC                              ;RETURN
001424  000000                  E1$:    HALT


```

```
                                        ;TEST WRITES AND READS DIFFERENT MASK
                                        ;OPTIONS ACCORDING TO SEGMENT RUN.
                                        ;HALTS ON ERROR

001426  013737  002022  172416  TEST2:  MOV     @#SELBUF,@#OPREG        ;SEL BUFFER
001434  013700  002026                  MOV     @#MASK, R0              ;MASK SELECT
001440  042700  177400                  BIC     #177400,R0              ;CLEAR UNUSED BITS
001444  010001                          MOV     R0,     R1              ;R1=SAVED MASK
001446  012702  177400                  MOV     #-256., R2              ;LOOP COUNTER
001452  062700  004400                  ADD     #4400,  R0              ;WRITE COMMAND
001456  010037  172416          W2$:    MOV     R0,     @#OPREG         ;WD
001462  105737  172414                  TSTB    @#CSR                   ;VALID WD
001466  100375                          BPL     .-4                     ;NO LOOP UNTIL
001470  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
001474  005202                          INC     R2
001476  001367                          BNE     W2$                     ;CONT UNTIL 0

001500  013737  002022  172416          MOV     @#SELBUF,@#OPREG        ;SEL BUFFER
001506  012700  177400                  MOV     #-256., R0              ;LOOP COUNTER
001512  012737  001400  172416  R2$:    MOV     #768.,  @#OPREG         ;RD
001520  105737  172414                  TSTB    @#CSR                   ;VALID RD
001524  100375                          BPL     .-4                     ;NO LOOP UNITL
001526  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
001532  013702  172416                  MOV     @#OPREG, R2             ;FETCH DATA
001536  042702  177400                  BIC     #177400, R2             ;CLEAR UNUSED BITS
001542  120102                          CMPB    R1,     R2              ;DATA=MASK
001544  001003                          BNE     E2$                     ;NO
001546  005200                          INC     R0
001550  001360                          BNE     R2$                     ;CONT UNTIL 0
001552  000207                          RTS     PC
001554  000000          E2$:    HALT                                    ;ERROR IN DATA READ
```

```
 1                                          ;SEGMENT READS AND WRITES RANDOM NUMBERS
 2                                          ;INTO SELECTED BUFFER HALTS ON  ERROR.
 3
 4 001556  004767  000134          TEST3:  JSR     PC,     GENRAN                  ;GENERATE # BLOCH
 5 001562  012704  002034                  MOV     #RANUM, R4                      ;ADDRESS OF BLOCK
 6 001566  012700  000400                  MOV     #256.,  R0                      ;NUMBER OF BYTES
 7 001572  013737  002022  172416          MOV     @#SELBUF, @#OPREG                       ;SEL BUFFER
 8 001600  012403                  W3$:    MOV     (R4)+,  R3                      ;RANDOM NUMBER
 9 001602  062703  004400                  ADI     #4400,  R3                      ;WD COMMAND
10 001606  010337  172416                  MOV     R3,     @#OPREG                 ;WD
11 001612  105737  172414                  TSTB    @#CSR                           ;GOOD WD
12 001616  100375                          BPL     .-4                             ;LOOP UNTIL
13 001620  105037  172414                  CLRB    @#CSR                           ;CLEAR
14 001624  077013                          SOB     R0,     W3$                     ;LOOP UNTIL 0
15 001626  013737  002022  172416          MOV     @#SELBUF,@#OPREG                ;SEL BUFFER
16 001634  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE RD
17 001642  105737  172414                  TSTB    @#CSR                           ;OK
18 001646  100375                          BPL     .-4                             ;LOOP UNTIL
19 001650  105037  172414                  CLRB    @#CSR
20 001654  012700  000400                  MOV     #256.,  R0                      ;COUNT
21 001660  012737  001400  172416  R3$:    MOV     #1400,  @#OPREG                 ;RD
22 001666  105737  172414                  TSTB    @#CSR                           ;READY
23 001672  100375                          BPL     .-4                             ;LOOP UNTIL
24 001674  013703  172416                  MOV     @#OPREG, R3                     ;FETCH BYTE
25 001700  042703  177400                  BIC     #177400, R3                     ;CLEAR MST
26 001704  024403                          CMP     -(R4),  R3                      ;DATA = DATA
27 001706  001002                          BNE     E3$                             ;NO ERROR
28 001710  077015                          SOB     R0,     R3$                     ;YES CONTINUE
29 001712  000207                          RTS     PC                              ;RETURN
30 001714  000000                  E3$:    HALT                                    ;ERROR IN TEST3
31
32                                         ;*******RANDOM NUMBER GENERATOR
33
34 001716  012703  002034          GENRAN: MOV     #RANUM, R3                      ;DATA BLOCK
35 001722  012704  177400                  MOV     #-256., R4                      ;GENERATE BY 256
36 001726  013700  002030          LP:     MOV     @#LONUM, R0                     ;SEED 1
37 001732  013701  002032                  MOV     @#HINUM, R1                     ;SEED 2
38 001736  012702  177771                  MOV     #-7,    R2                      ;COUNTER
39 001742  006300                  GR:     ASL     R0                              ;SHIFT
40 001744  006101                          ROL     R1                              ;ROTATE
41 001746  005202                          INC     R2                              ;INC AND CHECK
42 001750  001374                          BNE     GR
43 001752  063700  002030                  ADD     @#LONUM, R0                     ;MAKE IT X129
44 001756  005501                          ADC     R1                              ;PROP CARRY
45 001760  063701  002032                  ADD     @#HINUM, R1                     ;MAKE IT X129
46 001764  062700  001057                  ADD     #1057,  R0                      ;ADD CONSTANT
47 001770  005501                          ADC     R1                              ;PROP CARRY
48 001772  062701  047401                  ADD     #47401, R1                      ;ADD CONSTANT
49 001776  010037  002030                  MOV     R0,     @#LONUM                 ;SAVE
50 002002  010137  002032                  MOV     R1,     @#HINUM                 ;SAVE
51 002006  042701  177400                  BIC     #177400, R1                     ;CLEAR UNUSED BITS
52 002012  010123                          MOV     R1,     (R3)+                   ;STORE NUMBER
53 002014  005204                          INC     R4                              ;COUNT +1
54 002016  001343                          BNE     LP                              ;<>0 CONTINUE
55 002020  000207                          RTS     PC
56                                         ;******
```

AD-A078 392 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66158 SBIE-AD-E100 314 NL

2 OF 4
ADA 078392



```
                                        ;COMMON LATA

002022                          SELBUF: .BLKW
002024                          BFPCMD: .BLKW
002026                          MASK:   .BLKW
00203Ø  123456                  LONUM:  .WORD   123456
002032  176543                  HINUM:  .WORD   176543
002034                          RANUM:  .BLKW   256.

        001000                          .END CLEAR
```

```
BUFFER-DIAGNOSTICS      MACRO V03.01 27-MAR-79 18:01:51 PAGE 5-1
SYMBOL TABLE

BFPCMD  002024          E1$     001424          LONUM   002030          R2$     001512          TEST2   001426
CKBP    001356          E2$     001554          LP      001726          R3$     001660          TEST3   001556
CLEAR   001000          E3$     001714          MASK    002026          SELBUF  002022          TIME    001064
CLR$    001020          GENRAN  001716          OPREG = 172416          START   001066          W2$     001456
CSR   = 172414          GR      001742          RANUM   002034          TEST1   001252          W3$     001600
C1$     001324          HINUM   002032          RUN     001170

. ABS.  003034      000
        000000      001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:LIUBUF=DK:LIUBUF
```

3.1.6 LSI-11 REAL TIME CLOCK TEST

Source File: LSICLK.MAC

Task File: LSICLK.SAV

PURPOSE: To test LSI-11 RTC

DESCRIPTION: This program waits for interrupts from the real time clock and increments a counter when one is received. When the program does not detect an interrupt in a given time, it halts, detecting an error, and displays current program address.

```
                                        .TITLE  LSI-11 RTC DIAG.
                                        .IDENT  /V1.0/
000000                                  .ASECT
        001000                          .=      1000


                                        ;***********************************************
                                        ;* DIAGNOSTIC CHECKS (BEVENT L) CLOCK SIGNAL   *
                                        ;* AT 16.834 USEC USED AS THE RTC GENERATOR.   *
                                        ;* VECTORS TO LOCATION 100 FOR ADDRESS OF      *
                                        ;* HANDLER                                     *
                                        ;* NOTE: PROGRAM HALTS AT (ETO:) IF AN         *
                                        ;*       INTERRUPT FAILS TO BE RECEIVED.       *
                                        ;***********************************************


                                        ;CLEAR INTERRUPT AND VECTOR AREA
                                        ;SET LOCATION 100 TO TIMER

001000  000005                  CLEAR: RESET                                   ;RESET BUS
001002  012700  000340                 MOV     #340, R0                        ;PRI=7
001006  106400                         MTPS    R0
001010  012706  000500                 MOV     #500,   SP                      ;INITIALIZE STACK
001014  012702  000200                 MOV     #200,   R2                      ;R2=TOP OF VECTOR AREA
001020  005042                 CLR$:  CLR     -(R2)                           ;INSERT HALT INSTR.
Z 001022  010242                       MOV     R2,     -(R2)                   ;POINT TO HALT
001024  005702                         TST     R2                              ;LOC 0 YET
001026  001374                         BNE     CLR$                            ;NO CONTINUE
001030  012737  001000  000040         MOV     #CLEAR, @#40                    ;RESTART ADDRESS
001036  012700  000100                 MOV     #100,   R0                      ;VECTOR
001042  012710  001106                 MOV     #TIMER, (R0)                    ;ADDRESS OF HANDLER
001046  012701  000000                 MOV     #000,   R1                      ;CPU PRIOITY 2
001052  106401                         MTPS    R1                              ;SET CPU PRI.

                                        ;TIMEOUT LOOP: COUNTS UNTIL INTERRUPT
                                        ;IS RECEIVED, HALTS IF TIME-OUT COUNT
                                        ;REACHS LIMIT

001054  005000                 TIMOUT: CLR     R0                              ;CLEAR COUNT
001056  005001                         CLR     R1                              ;CLEAR LIMIT
001060  005200                 INCT:  INC     R0                              ;COUNT+1
001062  005700                         TST     R0                              ;OVER-FLOW YET?
001064  100375                         BPL     INCT                            ;NO CONTINUE
001066  000257                         CCC                                     ;CLEAR COND. CODES
001070  005201                         INC     R1                              ;COUNT+1
001072  023701  001160                 CMP     @#LIMIT,R1                      ;LIMIT YET
001076  001402                         BEQ     ETO                             ;YES HALT
001100  005000                         CLR     R0                              ;CLEAR COUNTER
001102  000766                         BR      INCT                            ;CONTINUE COUNTER
001104  000000                 ETO:   HALT                                    ;ERROR TIME-OUT
```

```
                                        ;INTERRUPT HANDLER FOR RTC
                                        
001106  005237  001162        TIMER:  INC     @#TIME                  ;TIME+1
001112  023737  001162  001164        CMP     @#TIME, @#OLDTIM        ;DID CLOCK ADVANCE
001120  001001                        BNE     T2$                     ;YES
001122  000000                        HALT                            ;NO-ERROR IN CLOCK
001124  013737  001162  001164  T2$:   MOV     @#TIME, @#OLDTIM        ;RESET OLDTIM
001132  022737  177777  001162        CMP     #177777, @#TIME         ;TIME LIMIT
001140  001004                        BNE     T3$                     ;NO RTI
001142  005037  001162                CLR     @#TIME                  ;YES CLEAR TIME
001146  005037  001164                CLR     @#OLDTIM                ;    CLEAR OLDTIM
001152  005000                  T3$:  CLR     R0                      ;CLEAR COUNT
001154  005001                        CLR     R1                      ;
001156  000002                        RTI                             ;RETURN FROM INTERRUPT
                                        
                                        ;DATA AREA
                                        
001160  000010                  LIMIT:  .WORD   10
001162  000000                  TIME:   .WORD   0
001164  000000                  OLDTIM: .WORD   0
                                        
                                        
        001000                          .END CLEAR
```

```
LSI-11 RTC DIAG.        MACRO V03.02B2-NOV-79 10:45:29 PAGE 2-1
SYMBOL TABLE

CLEAR   001000          INCT    001060          OLDTIM  001164          TIMER   001106          T2$     001124
CLR$    001020          LIMIT   001160          TIME    001162          TIMOUT  001054          T3$     001152
ETO     001104

. ABS.  001166      000
        000000      001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  56 PAGES
DK:LSICLK,DK:LSICLK=DK:LSICLK
```

3.1.7 MSCDM DIAGNOSTIC OPERATING INSTRUCTIONS

3.1.7.1 RUNNING DIAGNOSTICS ON PDU NODE

1. Insert diskette #1 into DX0:
2. Insert diagnostic diskette #16 into DX1:
3. Type .GET fid where fid=file name
4. Press BREAK KEY
5. Insure that the RTC is off
6. TYPE @1000G
7. Diagnostic will then loop until an error is detected whereby the current program address (PC) +2 will be displayed on the CRT. By looking at the source code for that diagnostic the address (PC) will indicate the error that was detected, and the user registers will hold the cause of the error.

3.1.7.2 RUNNING DIAGNOSTICS ON A LOOP NODE

1. Insert diskette #1 into DX0:
2. Insert diagnostic diskette #16 into DX1:
3. Turn the RTC off on node(s) under test
4. Run FDMLDR to load and start the diagnostic
5. Insure the RUN LED on the front panel for that node stays on (diagnostic running). Should the LED extinguish, an error was detected and the software has halted.

6. By attaching a CRT MAINTENANCE CABLE to the node and restarting program at address 1000 (@ 1000G); using the CRT the diagnostic will again begin to run and when the error develops again the (PC); will be displayed on the CRT. By looking at the source code for that diagnostic the address (PC); will indicate the error that was detected, and the user registers hold the cause of the error.

Note: Not all nodes contain a DLVII interface card that can support the CRT. Cards may have to be swapped with another node.

3.1.7.3 RUNNING DIAGNOSTICS ON SIG

SIG takes a special type of software which is in the .LDA format and when loaded it STARTS automatically. Also note that the SIGN loads via the asynchronous interface from node 26. The procedure other than that is the same for any other node on loop.

3.1.7.4 SEQUENCE OF DIAGNOSTICS

By running diagnostics in a certain sequence, the hardware error can be isolated down to a specific card. The recommended sequence is given below:

1. RUN LSICPU - CPU diagnostic
2. RUN LSIMEM - MEMORY diagnostic
3. RUN LIUINT - INTERFACE diagnostic
4. RUN LIURAM - ACRAM diagnostic
5. RUN LIUBUF - BUFFER diagnostic

3.2 IEEE PROGRAMS

3.2.1 IEEE Receive Program Description (IEEER)

The program initializes the LIU and then loops, waiting for packets from the loop. When a packet is received it is ACKed and printed on the LA36.

```
                                        .TITLE  IEEE RECEIVE
                                        .IDENT  /V1.0/
000000                                  .ASECT
          001000                        .=      1000

          172410                        BAR=    172410
          172412                        WCR=    172412
          172414                        CSR=    172414
          172416                        OPREG=  172416
          177560                        RCSR=   177560
          177562                        RBUF=   177562
          177564                        XCSR=   177564
          177566                        XBUF=   177566

          100000                        BIT15=  100000
          040000                        BIT14=  40000
          000400                        BIT08=  400
          000200                        BIT07=  200
          000100                        BIT06=  100
          000040                        BIT05=  40
          000020                        BIT04=  20
          000010                        BIT03=  10
          000004                        BIT02=  4
          000002                        BIT01=  2
          000001                        BIT00=  1

                                        ;CLEAR INTERRUPT AND TRAP VECTOR AREA

001000  000005                CLEAR:  RESET                           ;RESET BUS
001002  012700  000340                MOV     #340,   R0              ;CPU PRI=7
001006  106400                        MTPS    R0
001010  012706  000500                MOV     #500,   SP              ;INIT STACK POINTER
001014  012702  000200                MOV     #200,   R2              ;R2=TOP OF VECTOR AREA
001020  005042                CLR$:   CLR     -(R2)                   ;INSERT A HALT
Z 001022  010242                      MOV     R2,     -(R2)           ;POINT TO HALT
001024  005702                       TST     R2                      ;DONE ALL VECTORS
001026  001374                       BNE     CLR$                    ;NO CONTINUE
001030  012737  001000  000040       MOV     #CLEAR,@#40             ;RESTART ADDRESS
001036  012737  040100  172414       MOV     #40100, @#CSR           ;DISABLE INTERRUPTS(LIU)
001044  012700  000124               MOV     #124,   R0              ;LIU VECTOR ADDRESS
001050  012720  001720               MOV     #LIU,   (R0)+           ;POINT TO HANDLER
001054  012710  000340               MOV     #340,   (R0)            ;INTER. AT PRI=340
001060  012700  000102               MOV     #102,   R0              ;CLOCK VECTOR
001064  012710  000002               MOV     #2,     (R0)            ;INSERT A RTI
```

```
                                        ;SET AND RESET LINE SWITCHS
                                        
001070  012737  010420  172416  SWT:    MOV     #4368., @#OPREG         ;MODSTAT
001076  012737  004414  172416          MOV     #4414,  @#OPREG         ;SET LINE SWITCHS
001104  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
001110  100375                          BPL     .-4                     ;NO LOOP UNTIL
001112  105037  172414                  CLRB    @#CSR
001116  012737  010420  172416          MOV     #4368., @#OPREG         ;MODSTAT
001124  012737  004540  172416          MOV     #4540,  @#OPREG         ;RESET LINE SWITCHS
001132  105737  172414                  TSTB    @#CSR                   ;GOOD WD
001136  100375                          BPL     .-4                     ;NO LOOP UNTIL
001140  105037  172414                  CLRB    @#CSR
                                        
                                        ;ACRAM ROUTINE
                                        
001144  012737  010402  172416  RAM:    MOV     #4354., @#OPREG         ;LDADR
001152  012737  004400  172416          MOV     #2304., @#OPREG         ;ADDR=0 (WD)
001160  105737  172414                  TSTB    @#CSR                   ;GOOD WD
001164  100375                          BPL     .-4                     ;NO LOOP UNTIL
001166  105037  172414                  CLRB    @#CSR
001172  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
001200  012700  177400                  MOV     #-256., R0              ;COUNTER
001204  012737  004407  172416  R2$:    MOV     #2311., @#OPREG         ;NULL WD
001212  105737  172414                  TSTB    @#CSR                   ;GOOD WD
001216  100375                          BPL     .-4                     ;NO LOOP UNTIL
001220  105037  172414                  CLRB    @#CSR
001224  005200                          INC     R0                      ;COUNT+1
001226  001366                          BNE     R2$
001230  012737  010402  172416          MOV     #4354., @#OPREG         ;LDADR DREAD(ADDR)
001236  013700  002530                  MOV     @#NODE, R0              ;SEL NODE
001242  062700  004400                  ADD     #2304., R0              ;ADD WD COMMAND
001246  010037  172416                  MOV     R0,     @#OPREG         ;WD
001252  105737  172414                  TSTB    @#CSR                   ;GOOD WD
001256  100375                          BPL     .-4                     ;NO LOOP UNTIL
001260  105037  172414                  CLRB    @#CSR
001264  012737  010401  172416  R4$:    MOV     #4353., @#OPREG         ;SEL ACRAM
001272  012737  004404  172416          MOV     #2308., @#OPREG         ;DREAD BITS
001300  105737  172414                  TSTB    @#CSR                   ;GOOD WD
001304  100375                          BPL     .-4                     ;NO LOOP UNTIL
001306  105037  172414                  CLRB    @#CSR
                                        
                                        ;INITIALIZE BUFFERS
                                        
001312  012737  010410  002534  INBUF0: MOV     #10410, @#BUFCMD        ;RDBUFADR COMMAND
001320  012737  010440  002532          MOV     #10440, @#SELBUF        ;SEL INBUF0
001326  004767  000122                  JSR     PC,     ZEROBP          ;SET POINTER=0
001332  012737  001400  172416          MOV     #1400,  @#OPREG         ;POINTER=FF
001340  105737  172414                  TSTB    @#CSR                   ;GOOD RD
001344  100375                          BPL     .-4                     ;NO LOOP UNTIL
001346  105037  172414                  CLRB    @#CSR
001352  012737  010610  002534  INBUF1: MOV     #10610, @#BUFCMD        ;RDBUFADR COMMAND
001360  012737  010640  002532          MOV     #10640, @#SELBUF        ;SEL INBUF1 COMMAND
001366  004767  000062                  JSR     PC,     ZEROBP          ;POINTER=0
001372  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE RD POINTER=FF
001400  105737  172414                  TSTB    @#CSR                   ;GOOD RD
001404  100375                          BPL     .-4                     ;NO LOOP UNTIL
001406  105037  172414                  CLRB    @#CSR
```

```
     58 001412  012737  010510  002534  OTBUF0: MOV     #10510, @#BUFCMD        ;RDBUFADR COMMAND
     59 001420  012737  010540  002532          MOV     #10540, @#SELBUF        ;SEL OUTBUF0
     60 001426  004767  000022                  JSR     PC,     ZEROBP          ;POINTER=0
     61 001432  012737  010710  002534  OTBUF1: MOV     #10710, @#BUFCMD        ;RDBUFADR COMMAND
     62 001440  012737  010740  002532          MOV     #10740, @#SELBUF        ;SEL OUTBUF1 COMMAND
     63 001446  004767  000002                  JSR     PC,     ZEROBP
     64 001452  000466                          BR      STAT                    ;RESET COMPLETE
     65
```

```
 1 001454  013700  002532          ZEROBP: MOV     @#SELBUF, R0            ;SEL BUFFER COMMAND
 2 001460  013701  002534                  MOV     @#BUFCMD, R1            ;RDBUFADR COMMAND
 3 001464  010137  172416                  MOV     R1,       @#OPREG       ;RDBUFADR
 4 001470  012737  001400  172416         MOV     #1400,    @#OPREG       ;RD
 5 001476  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 6 001502  100375                          BPL     .-4                     ;NO LOOP UNTIL
 7 001504  105037  172414                  CLRB    @#CSR
 8 001510  013702  172416                  MOV     @#OPREG, R2             ;FETCH POINTER
 9 001514  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
10 001520  010037  172416                  MOV     R0,       @#OPREG       ;SEL BUFFER
11 001524  022702  000000          CMP:    CMP     #0,       R2            ;POINTER=0?
12 001530  001412                          BEQ     RECK                    ;YES RECHECK
13 001532  012737  001400  172416         MOV     #1400,    @#OPREG       ;NO FALSE RD
14 001540  105737  172414                  TSTB    @#CSR                   ;GOOD RD
15 001544  100375                          BPL     .-4                     ;NO LOOP UNTIL
16 001546  105037  172414                  CLRB    @#CSR
17 001552  005302                          DEC     R2                      ;LOCAL POINTER-1
18 001554  000763                          BR      CMP                     ;BR UNTIL 0
19 001556  010137  172416          RECK:   MOV     R1,       @#OPREG       ;RDBUFADR
20 001562  012737  001400  172416         MOV     #1400,    @#OPREG       ;RD
21 001570  105737  172414                  TSTB    @#CSR                   ;GOOD RD
22 001574  100375                          BPL     .-4                     ;NO LOOP UNTIL
23 001576  105037  172414                  CLRB    @#CSR
24 001602  013702  172416                  MOV     @#OPREG, R2             ;FETCH NEW POINTER
25 001606  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
26 001612  022702  000000                  CMP     #0,       R2            ;POINTER=0
27 001616  001401                          BEQ     RTNB                    ;YES RETURN
28 001620  000000                          HALT                            ;ERROR IN BUFFER POINTER
29 001622  010037  172416          RTNB:   MOV     R0,       @#OPREG       ;SEL BUFFER
30 001626  000207                          RTS     PC                      ;RETURN
31
32                                         ;READ AND CLEAR STATUS
33
34 001630  012737  010400  172416  STAT:   MOV     #4352.,   @#OPREG       ;WCR:RS
35 001636  012737  002400  172416          MOV     #1280.,   @#OPREG       ;RS(FALSE)
36 001644  012737  002400  172416          MOV     #1280.,   @#OPREG       ;RS
37 001652  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
38 001656  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
39 001662  022700  000000                  CMP     #0,       R0            ;STATUS=0 ?
40 001666  001003                          BNE     ESTAT                   ;NO STOP PROGRAM
41 001670  005037  172414                  CLR     @#CSR                   ;YES CLEAR BLIUI CSR
42 001674  000401                          BR      LSLP                    ;START TEST
43 001676  000000                  ESTAT:  HALT                            ;ERROR IN LIU STATUS
44
45
```

```
 1                                                ;LOWER PRIORITY AND LOOP FOR INTERRUPT
 2
 3 001700  012700  000000          LSLP:  MOV     #000,   R0                    ;CPU PRI=0
 4 001704  106400                         MTPS    R0
 5 001706  052737  040000  172414        BIS     #BIT14, @#CSR                 ;ENABLE INTERRUPTS
 6 001714  000777                         BR                                    ;LOOP
 7 001716  000000                         HALT                                  ;PROGRAM ERROR
 8
 9                                                ;LIU INTERRUPT HANDLER
10
11 001720  010046                 LIU:    MOV     R0,     -(SP)                 ;SAVE R0
12 001722  010146                         MOV     R1,     -(SP)                 ;SAVE R1
13 001724  010246                         MOV     R2,     -(SP)                 ;SAVE R2
14 001726  010346                         MOV     R3,     -(SP)                 ;SAVE R3
15 001730  012737  010400  172416        MOV     #4352., @#OPREG               ;READ STATUS 1
16 001736  012737  002400  172416        MOV     #1280., @#OPREG               ;RS
17 001744  013700  172416                MOV     @#OFREG, R0
18 001750  042700  177400                BIC     #177400, R0                   ;FETCH & CLEAR MST BYTE
19
20                                               ;FIND WHY IT INTERRUPTED
21
22 001754  132700  000004        BFOV0:  BITB    #BIT02, R0                    ;DID INBUF0 OVER-FLOW
23 001760  001401                         BEQ     BFOV1                         ;NO
24 001762  000000                         HALT                                  ;YES STOP PROGRAM
25 001764  132700  000010        BFOV1:  BITB    #BIT03, R0                    ;DID INBUF1 OVER-FLOW
26 001770  001401                         BEQ     PRSW                          ;NO
27 001772  000000                         HALT                                  ;YES STOP PROGRAM
28 001774  132700  000020        PRSW:   BITB    #BIT04, R0                    ;DID PRIMARY GO DOWN
29 002000  001402                         BEQ     BKSW                          ;NO
30 002002  000167  000450                 JMP     DONE                          ;YES ENABLE INTERRUPTS
31 002006  132700  000040        BKSW:   BITB    #BIT05, R0                    ;DID BACKUP GO DOWN
32 002012  001402                         BEQ     WTDT                          ;NO
33 002014  000167  000436                 JMP     DONE                          ;YES ENABLE INTERRUPTS
34 002020  132700  000100        WTDT:   BITB    #BIT06, R0                    ;DID WRT-TOKEN DETECT
35 002024  001401                         BEQ     LATCH                         ;NO
36 002026  000000                         HALT                                  ;YES STOP PROGRAM
37 002030  132700  000200        LATCH:  BITB    #BIT07, R0                    ;DID LATCH GET SET
38 002034  001402                         BEQ     IE0F                          ;NO
39 002036  000167  000414                 JMP     DONE                          ;YES ENABLE INTERRUPTS
40
```

```
                                        ;INPUT BUFFER 0
                                        ;
002042  132700  000001        IB0F:   BITB    #BIT00, R0              ;DID INPUT BUF 0 FILL
002046  001501                        BEQ     IB1F                    ;NO
002050  012737  010410  172416        MOV     #4360., @#OPREG         ;YES READ POINTER
002056  012737  001400  172416        MOV     #768.,  @#OPREG         ;RD
002064  105737  172414                TSTB    @#CSR                   ;READY
002070  100375                        BPL     .-4                     ;NO LOOP UNTIL
002072  105037  172414                CLRB    @#CSR
002076  013703  172416                MOV     @#OPREG,  R3            ;FETCH DATA
002102  042703  177400                BIC     #177400, R3             ;CLEAR MST BYTE
002106  012737  010440  172416        MOV     #4384.,  @#OPREG        ;SEL INBF0
002114  012737  001400  172416        MOV     #768.,  @#OPREG         ;FALSE READ
002122  105737  172414                TSTB    @#CSR                   ;VALID RD
002126  100375                        BPL     .-4                     ;NO LOOP UNTIL
002130  105037  172414                CLRB    @#CSR
002134  005403                        NEG     R3                      ;2'S COMP. OF COUNT
002136  010337  172412                MOV     R3,     @#WCR           ;BYTE COUNT
002142  012737  002536  172410        MOV     #DATA,  @#BAR           ;DMA TO LOC=5000
002150  012737  021000  172416        MOV     #8704., @#OPREG         ;FIRE DMA
002156  000240                        NOP                             ;DELAY
002160  105737  172414                TSTB    @#CSR                   ;GOOD DMA
002164  100375                        BPL     .-4
002166  105037  172414                CLRB    @#CSR
002172  012737  004400  172416  PB0$:   MOV     #2304., @#OPREG         ;FALSE WD
002200  105737  172414                TSTB    @#CSR                   ;VALID WD
002204  100375                        BPL     .-4                     ;NO LOOP UNTIL
002206  105037  172414                CLRB    @#CSR
002212  004767  000264                JSR     PC,     PRTPK           ;PRINT IT
002216  012737  010600  172416        MOV     #4480., @#OPREG         ;READ STATUS 1
002224  012737  002400  172416        MOV     #1280., @#OPREG         ;RS
002232  013702  172416                MOV     @#OPREG, R2             ;FETCH DATA
002236  042702  177400                BIC     #177400,  R2            ;CLEAR MST BYTE
002242  132702  000001                BITB    #BIT00,  R2             ;CRC GOOD
002246  001001                        BNE     IB1F                    ;YES CONTINUE
002250  000000                ECRC0:  HALT                            ;ERROR BAD PACKET
                                        ;
                                        ;INPUT BUFFER 1
                                        ;
002252  132700  000002        IB1F:   BITB    #BIT01, R0              ;DID INBUF1 FILL
002256  001477                        BEQ     DONE                    ;ALL DONE
002260  012737  010610  172416        MOV     #4488., @#OPREG         ;RDBUF POINTER
002266  012737  001400  172416        MOV     #768.,  @#OPREG         ;RD
002274  105737  172414                TSTB    @#CSR                   ;READY
002300  100375                        BPL     .-4                     ;NO LOOP UNTIL
002302  105037  172414                CLRB    @#CSR
002306  013703  172416                MOV     @#OPREG, R3             ;FETCH DATA
002312  042703  177400                BIC     #177400, R3             ;CLEAR MST BYTE
002316  012737  010640  172416        MOV     #4512.,  @#OPREG        ;SEL INBUF1
002324  012737  001400  172416        MOV     #768.,  @#OPREG         ;FALSE READ
002332  105737  172414                TSTB    @#CSR                   ;VALID RD
002336  100375                        BPL     .-4                     ;NO LOOP UNTIL
002340  105037  172414                CLRB    @#CSR
002344  005403                        NEG     R3                      ;2'S COMP OF BYTE COUNT
002346  010337  172412                MOV     R3,     @#WCR           ;BYTE COUNT
002352  012737  002536  172410        MOV     #DATA,  @#BAR           ;WRITE TO LOC=6000
002360  012737  021000  172416        MOV     #8704., @#OPREG         ;FIRE DMA
```

```
    58 002366  000240                          NOP                             ;DELAY
    59 002370  105737  172414                   TSTB    @#CSR                   ;?GOOD DMA
    60 002374  100375                          BPL     .-4
    61 002376  012737  004400  172416  PB1$:   MOV     #2304., @#OPREG         ;FALSE WD
    62 002404  105737  172414                   TSTB    @#CSR                   ;VALID WD
    63 002410  100375                          BPL     .-4                     ;NO LOOP UNTIL
    64 002412  105037  172414                   CLRB    @#CSR
    65 002416  004767  000060                   JSR     PC,     PRTPK           ;PRINT IT
    66 002422  012737  010600  172416           MOV     #4480., @#OPREG         ;READ STATUS 1
    67 002430  012737  002400  172416           MOV     #1280., @#OPREG         ;RS
    68 002436  013702  172416                   MOV     @#OPREG, R2             ;FETCH DATA
    69 002442  042702  177400                   BIC     #177400, R2             ;CLEAR MST BYTE
    70 002446  132702  000002                   BITB    #BIT01, R2              ;GOOD CRC
    71 002452  001001                          BNE     DONE                    ;CRC OK
    72 002454  000000                  ECRC1:  HALT                            ;BAD CRC INBUF1
    73
    74 002456  012603                  DONE:   MOV     (SP)+,  R3              ;RESTORE R3
    75 002460  012602                          MOV     (SP)+,  R2              ;RESTORE R2
    76 002462  012601                          MOV     (SP)+,  R1              ;RESTORE R1
    77 002464  012600                          MOV     (SP)+,  R0              ;RESTORE R0
    78 002466  005037  172414                   CLR     @#CSR                   ;CLEAR BLIUI STATUS
    79 002472  052737  040000  172414           BIS     #BIT14, @#CSR           ;ENABLE BLIUI
    80 002500  000002                          RTI
    81
    82
    83 002502  012705  002536          PRTPK:  MOV     #DATA,  R5              ;ADDRESS OF PACKET
    84 002506  105737  177564          P$:     TSTB    @#XCSR                  ;PORT READY
    85 002512  100375                          BPL     .-4                     ;NO LOOP
    86 002514  112537  177566                   MOVB    (R5)+,@#XBUF            ;PRINT BYTE
    87 002520  122715  000003                   CMPB    #003,   (R5)            ;ETX YET
    88 002524  001370                          BNE     P$
    89 002526  000207                          RTS     PC
    90
    91                                         ;DATA BLOCK
    92
    93 002530     014                  NODE:   .BYTE   014
    94 002531     000                          .BYTE   00
    95 002532                          SELBUF: .BLKW
    96 002534                          BUFCMD: .BLKW
    97 002536                          DATA:   .BLKB   25.
    98 002567     000                          .BYTE   0
    99
   100          001000                          .END    CLEAR
```

```
IEEE RECEIVE   MACRO V03.01 27-MAR-79 19:50:30 PAGE 5-2
SYMBOL TABLE

BAR    = 172410          BIT14  = 040000          ECRC1   002454          OTBUF0  001412          RTNB    001622
BFOV0    001754          BIT15  = 100000          ESTAT   001676          OTBUF1  001432          R2$     001204
BFOV1    001764          BKSW     002006          IB0F    002042          PB0$    002172          R4$     001264
BIT00  = 000001          BUFCMD   002534          IB1F    002252          PB1$    002376          SELBUF  002532
BIT01  = 000002          CLEAR    001000          INBUF0  001312          PRSW    001774          STAT    001630
BIT02  = 000004          CLR$     001020          INBUF1  001352          PRTPK   002502          SWT     001070
BIT03  = 000010          CMP      001524          LATCH   002030          P$      002506          WCR    = 172412
BIT04  = 000020          CSR    = 172414          LIU     001720          RAM     001144          WTDT    002020
BIT05  = 000040          DATA     002536          LSLP    001700          RBUF   = 177562          XBUF   = 177566
BIT06  = 000100          DONE     002456          NODE    002530          RCSR   = 177560          XCSR   = 177564
BIT07  = 000200          ECRC0    002250          OPREG  = 172416          RECK    001556          ZEROBP  001454
BIT08  = 000400

. ABS.  002570     000
        000000     001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:RIEEE=DK:RIEEE
```

3.2.2 IEEE Send Program Description (IEEES)

The program initializes the LIU and IEEE interface, once initialized, the program reads data from the IEEE interface and writes it to the loop.

```
                                        .TITLE IEEE SEND
                                        .IDENT  /V1.0/
000000                                  .ASECT
001000                                  .=      1000

172410                                  BAR=    172410
172412                                  WCR=    172412
172414                                  CSR=    172414
172416                                  OPREG=  172416
177560                                  ICB=    177560
177562                                  IDB=    177562


                                        ;CLEAR INTERRUPT AND TRAP VECTOR AREA

001000  000005                CLEAR:    RESET                           ;RESET BUS
001002  012700  000340                  MOV     #340,   R0              ;CPU PRI=7
001006  106400                          MTPS    R0
001010  012706  000500                  MOV     #500,   SP              ;INITIALIZE STACK
001014  012702  000200                  MOV     #200,   R2              ;R2=TOP VECTOR AREA
001020  005042                CLR$:     CLR     -(R2)                   ;INSERT A HALT
001022  010242                          MOV     R2,     -(R2)           ;POINT TO HALT
001024  005702                          TST     R2                      ;ALL VECTORS YET
001026  001374                          BNE     CLR$
001030  012700  000102                  MOV     #102,   R0              ;CLOCK VECTOR
001034  012710  000002                  MOV     #2,     (R0)            ;INSERT A RTI
001040  012737  001000  000040          MOV     #CLEAR, @#40            ;RESETART ADDRESS
001046  012737  001642  000434          MOV     #IEEE,  @#434           ;VECTOR FO IEEE
001054  012737  000200  000436          MOV     #200,   @#436           ;PRI=5
001062  012737  040100  172414          MOV     #040100, @#CSR          ;DISABLE LIU INTERRUPTS

                                        ;SET AND RESET LINE SWITCHS

001070  012737  010420  172416  SWT:    MOV     #4368., @#OPREG         ;MODSTAT
001076  012737  004414  172416          MOV     #4414,  @#OPREG         ;SET LINE SWITCHS
001104  105737  172414                  TSTB    @#CSR                   ;GOOD WD
001110  100375                          BPL     .-4                     ;NO LOOP UNTIL
001112  105037  172414                  CLRB    @#CSR
001116  012737  010420  172416  S1$:    MOV     #4368., @#OPREG         ;MODSTAT
001124  012737  004540  172416          MOV     #4540,  @#OPREG         ;RESET LINE SWITCHS
001132  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
001136  100375                          BPL     .-4                     ;NO LOOP UNTIL
001140  105037  172414                  CLRB    @#CSR
```

```
                                        ;INITAILIZE I/O BUFFERS
                                        
001144  012737  010410  002016  INBUF0: MOV     #10410, @#BUFCMD        ;RDBUFADR COMMAND
001152  012737  010440  002014          MOV     #10440, @#SELBUF        ;SEL INBUF0 COMMAND
001160  004767  000122                  JSR     PC,     ZEROBP          ;SET POINTER=0
001164  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ POINTER=FF
001172  105737  172414                  TSTB    @#CSR
001176  100375                          BPL     .-4                     ;NO LOOP UNTIL
001200  105037  172414                  CLRB    @#CSR
001204  012737  010610  002016  INBUF1: MOV     #10610, @#BUFCMD        ;RDBUFADR COMMAND
001212  012737  010640  002014          MOV     #10640, @#SELBUF        ;SEL INBUF1
001220  004767  000062                  JSR     PC,     ZEROBP          ;SET POINTER=0
001224  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ POINTER=FF
001232  105737  172414                  TSTB    @#CSR                   ;GOOD RD
001236  100375                          BPL     .-4                     ;NO LOOP UNTIL
001240  105037  172414                  CLRB    @#CSR
001244  012737  010510  002016  OTBUF0: MOV     #10510, @#BUFCMD        ;RDBUFADR COMMAND
001252  012737  010540  002014          MOV     #10540, @#SELBUF        ;SEL OUTBUF0 COMMAND
001260  004767  000022                  JSR     PC,     ZEROBP          ;SET POINTER=0
001264  012737  010710  002016  OTBUF1: MOV     #10710, @#BUFCMD        ;RDBUFADR COMMAND
001272  012737  010740  002014          MOV     #10740, @#SELBUF        ;SEL OUTBUF1
001300  004767  000002                  JSR     PC,     ZEROBP          ;SET POINTER=0
001304  000520                          BR      STAT                    ;RESET COMPLETE
                                        
001306  013700  002014          ZEROBP: MOV     @#SELBUF, R0            ;SEL BUFFER COMMAND
001312  013701  002016                  MOV     @#BUFCMD, R1            ;RDBUFADR COMMAND
001316  010137  172416                  MOV     R1,     @#OPREG         ;RDBUFADR
001322  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
001330  105737  172414                  TSTB    @#CSR                   ;GOOD RD
001334  100375                          BPL     .-4                     ;NO LOOP UNTIL
001336  105037  172414                  CLRB    @#CSR
001342  013702  172416                  MOV     @#OPREG, R2             ;FETCH POINTER
001346  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
001352  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
001356  022702  000000          CMP:    CMP     #0,     R2              ;POINTER=0
001362  001412                          BEQ     RECK                    ;YES RECHECK
001364  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO DO A (RD)
001372  105737  172414                  TSTB    @#CSR                   ;GOOD RD
001376  100375                          BPL     .-4                     ;MO LOOP UNTIL
001400  105037  172414                  CLRB    @#CSR
001404  005302                          DEC     R2                      ;LOCAL POINTER-1
001406  000763                          BR      CMP                     ;BR UNTIL 0
001410  010137  172416          RECK:   MOV     R1,     @#OPREG         ;RDBUFADR
001414  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
001422  105737  172414                  TSTB    @#CSR                   ;GOOD RD
001426  100375                          BPL     .-4                     ;NO LOOP UNTIL
001430  105037  172414                  CLRB    @#CSR
001434  013702  172416                  MOV     @#OPREG, R2             ;FETCH POINTER
001440  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
001444  022702  000000                  CMP     #0,     R2              ;POINTER=0
001450  001401                          BEQ     RTN                     ;YES SEL BUFFER & RETURN
001452  000000                          HALT                            ;ERROR IN BUF POINTER
001454  010037  172416          RTN:    MOV     R0,     @#OPREG         ;SEL BUFFER
001460  000207                          RTS     PC                      ;RETURN
```

```
 1
 2                                          ;INITIALIZE ACRAM
 3
 4 001462  012737  010402  172416           MOV     #4354., @#OPREG          ;LDACR
 5 001470  012737  004400  172416           MOV     #2304., @#OPREG          ;ADD=0
 6 001476  105737  172414                   TSTB    @#CSR                    ;GOOD WD
 7 001502  100375                           BPL     .-4                      ;LOOP UNTIL
 8 001504  105037  172414                   CLRB    @#CSR                    ;CLEAR DONE
 9 001510  012737  010401  172416           MOV     #4353., @#OPREG          ;SEL ACRAM
10 001516  012700  177400                   MOV     #-256., R0               ;ALL ADDRESS
11 001522  012737  004407  172416  1$:      MOV     #2311., @#OPREG          ;NULL
12 001530  105737  172414                   TSTB    @#CSR                    ;GOOD WD
13 001534  100375                           BPL     .-4                      ;LOOP UNTIL
14 001536  105037  172414                   CLRB    @#CSR                    ;CLEAR DONE
15 001542  005200                           INC     R0
16 001544  001366                           BNE     1$
17
18                                          ;READ AND CLEAR STATUS
19
20 001546  012737  010400  172416  STAT:    MOV     #4352., @#OPREG          ;RS(0)
21 001554  012737  002400  172416           MOV     #1280., @#OPREG          ;READ FALSE STATUS
22 001562  012737  002400  172416           MOV     #1280., @#OPREG          ;RS
23 001570  013700  172416                   MOV     @#OPREG, R0              ;FETCH STATUS
24 001574  042700  177400                   BIC     #177400, R0              ;CLEAR MST BYTE
25 001600  022700  000000                   CMP     #0,      R0              ;STATUS=0
26 001604  001003                           BNE     ESTAT                    ;NO ERROR STOP PROGRAM
27 001606  005037  172414                   CLR     @#CSR                    ;CLEAR BLIUI CSR
28 001612  000401                           BR      START                    ; AND CONTINUE
29 001614  000000                ESTAT:     HALT                             ;ERROR IN STATUS
30
```

```
                                        ;IEEE FETCH AND SEND
                                        
001616  012700  002020        START:  MOV     #DATA,  R0              ;DATA BLOCK
001622  005001                        CLR     R1                      ;DATA COUNTER
001624  012700  000000                MOV     #000,   R0              ;PRI=0
001630  106400                        MTPS    R0                      ;LOWER CPU
001632  012737  000320  177560        MOV     #320,   @#ICB           ;FIRE IEEE BUS OK
001640  000777                WAIT:   BR      .                       ;LOOP UNTIL READY
                                        
001642  113705  177562        IEEE:   MOVB    @#IDB,  R5              ;FETCH BYTE
001646  042705  177400                BIC     #177400,R5              ;CLEAR MST
001652  110520                        MOVB    R5,(R0)+                ;SAVE BYTE
001654  005201                        INC     R1                      ;COUNT +1
001656  022701  000015                CMP     #13.,   R1              ;MESS. ALL REV YET
001662  001005                        BNE     ERTN                    ;NO DAC
001664  004767  000014                JSR     PC,     SEND            ;SEND IT ON THE LOOP
001670  012700  002020                MOV     #DATA,  R0              ;RESET STACK
001674  005001                        CLR     R1                      ;RESET POINTER
001676  005037  177562        ERTN:   CLR     @#IDB                   ;DAC
001702  000002                        RTI
                                        
                                        ;DMA DATA BLOCK TO OUTBUF0
                                        
001704  012737  010540  172416  SEND:   MOV     #4448., @#OPREG         ;SEL OUTBUF0
001712  012737  002020  172410          MOV     #DATA,  @#BAR           ;BUS ADDRESS
001720  012700  000026                MOV     #22.,   R0              ;BYTE COUNT
001724  005400                        NEG     R0                      ;2'COMP OF BC
001726  010037  172412                MOV     R0,     @#WCR           ;WRITE IT
001732  012737  024000  172416        MOV     #10240., @#OPREG        ;FIRE DMA
001740  000240                        NOP                             ;DELAY
001742  105737  172414                TSTB    @#CSR                   ;GOOD DMA
001746  100401                        BMI     SBITS                   ;YES SET WRITE BITS
001750  000000                        HALT                            ;ERROR BAD DMA
                                        
                                        ;SET OBUF0F/WRTCMD BITS
                                        
001752  012737  010420  172416  SBITS:  MOV     #4368., @#OPREG         ;MODSTAT
001760  012737  004421  172416          MOV     #4421,  @#OPREG         ;SET BUFFER FULL/WTCMD
001766  105737  172414                TSTB    @#CSR                   ;GOOD WRITE?
001772  100375                        BPL     .-4                     ;NO LOOP UNTIL
001774  105037  172414                CLRB    @#CSR
002000  012700  077777                MOV     #077777, R0             ;DELAY
002004  000240                DELAY:  NOP
002006  077002                        SOB     R0,     DELAY           ;LOOP ON DELAY
002010  000207                        RTS     PC
                                        
                                        ;DATA BLOCK
                                        
002012  125125                MASK:   .WORD   125125
002014                        SELBUF: .BLKW
002016                        BUFCMD: .BLKW
002020                        DATA:   .BLKB   20.
002044     003                        .BYTE   003
002045     014                        .BYTE   014
                                        
        001000                      .END    CLEAR
```

```
IEEE SEND       MACRO V03.01 27-MAR-79 19:51:15 PAGE 4-1
SYMBOL TABLE

BAR    = 172410          DELAY   002004          INBUF0  001144          RECK    001410          STAT    001546
BUFCMD  002016           ERTN    001676          INBUF1  001204          RTN     001454          SWT     001070
CLEAR   001000           ESTAT   001614          MASK    002012          SBITS   001752          S1$     001116
CLR$    001020           ICB    = 177560         OPREG  = 172416         SELBUF  002014          WAIT    001640
CMP     001356           IDB    = 177562         OTBUF0  001244          SEND    001704          WCR    = 172412
CSR    = 172414          IEEE    001642          OTBUF1  001264          START   001616          ZEROBP  001306
DATA    002020

. ABS.  002046      000
        000000      001
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:SIEEE=DK:SIEEE
```

3.3 LOADER PROM DESCRIPTION

The prom loading programs for MSCDM are written in DEC MARCRO-11 Assembly language and the object code used to burn the loading proms. This allows a node attached to the loop to be loaded on power-up from the local host processor (PDP-11/V03). Nodes 21, 22, 23, 27 and 28 all have identical loader proms except for their individual read addresses. Node 25 proms also has a subroutine which allows packets received from the loop to be printed on its local printer (LA-36). This subroutine is used by the PGLOOP program. Node 26 proms also has a subroutine which allows packet received from loop destine for the SIG is passed over the local 9600 baud interface attached to the SIG. This subroutine is used by the FDMLDR program. The SIG has a special type of loader prom which load from a 9600 baud interface. This loader checks CRC on memory bytes and also automatically start the loader program.

```
                                        .TITLE PROM NODE 21
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000

        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1

173000  000005                  START:  RESET                   ;RESET BUS
173002  012700  000340                  MOV     #340,   R0      ;PRI=7
173006  106400                          MTPS    R0              ;CPU PRI
173010  012706  160500                  MOV     #160500, SP     ;RESET STACK

                                        ;INITIALIZE LINE SWITCHS

173014  012737  010420  172416  INITLS: MOV     #4368., @#OPREG ;MOD STAT
173022  012737  004414  172416          MOV     #4414,  @#OPREG ;SET SWITCHES
173030  105737  172414                  TSTB    @#CSR           ;GOOD WD
173034  100375                          BPL     .-4             ;LOOP UNTIL
173036  105037  172414                  CLRB    @#CSR
173042  012737  004540  172416          MOV     #4540,  @#OPREG ;RESET SWITCHS
173050  105737  172414                  TSTB    @#CSR           ;GOOD WD
173054  100375                          BPL     .-4             ;LOOP UNTIL
173056  105037  172414                  CLRB    @#CSR
173062  013703  174652                  MOV     @#STATUS, R3    ;LO CATION OF STATUS
173066  005005                          CLR     R5              ;CLEAR ERROR COUNT
```

```
                                        ;INITIALIZE LIU
                                        
173070  005004                  INIT:   CLR     R4                      ;CLEAR LOOP COUNT
173072  012737  010402  172416          MOV     #4354., @#OPREG         ;LDADR
173100  012737  004400  172416          MOV     #2304., @#OPREG         ;ADDRESS=0
173106  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173112  100375                          BPL     .-4                     ;NO RETRY
173114  105037  172414                  CLRB    @#CSR
173120  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
173126  012700  000400                  MOV     #256., R0               ;COUNTER
173132  012737  004407  172416  1$:     MOV     #2311., @#OPREG         ;WRITE A NULL
173140  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173144  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173146  105037  172414                  CLRB    @#CSR
173152  077011                          SOB     R0,     1$              ;LOOP UNTIL ZERO
173154  012700  000400                  MOV     #256.,  R0              ;COUNTER
173160  012737  001400  172416  2$:     MOV     #768.,  @#OPREG         ;RD
173166  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173172  100375                          BPL     .-4                     ;NO RETRY
173174  105037  172414                  CLRB    @#CSR
173200  013701  172416                  MOV     @#IOBUF, R1             ;FETCH DATA
173204  042701  177760                  BIC     #177760, R1             ;CLEAR BITS
173210  022701  000007                  CMP     #7,      R1             ;DATA=NULL
173214  001401                          BEQ     3$                      ;YES
173216  005204                          INC     R4                      ;NO REPORT IT
173220  077021                  3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO
173222  012737  010402  172416  STRAM:  MOV     #4354., @#OPREG         ;LDCTADR
173230  013737  174662  172416          MOV     @#NODE, @#OPREG         ;SEND ADDRESS
173236  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
173242  100375                          BPL     .-4                     ;LOOP UNTIL
173244  105037  172414                  CLRB    @#CSR
173250  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
173256  012737  004404  172416          MOV     #4404,  @#OPREG         ;LOAD A 4
173264  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
173270  100375                          BPL     .-4                     ;NO LOOP UNTIL
173272  105037  172414                  CLRB    @#CSR
173276  012701  010410          INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
173302  012700  010440                  MOV     #10440, R0              ;SEL INBUFO
173306  004767  000106                  JSR     PC,     ZEROBP          ;SET POINTER=0
173312  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
173320  105737  172414                  TSTB    @#CSR                   ;GOOD READ
173324  100375                          BPL     .-4                     ;NO RETRY
173326  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173332  012701  010610          INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
173336  012700  010640                  MOV     #10640, R0              ;SEL INBUF1 COMMAND
173342  004767  000052                  JSR     PC,     ZEROBP          ;POINTER=0
173346  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
173354  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173360  100375                          BPL     .-4                     ;NO RETRY
173362  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173366  012701  010510          OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
173372  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
173376  004767  000016                  JSR     PC,     ZEROBP
173402  012701  010710          OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
173406  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
173412  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
173416  000462                          BR      STAT                    ;GO CLEAR STATUS
```

```
                                        ;INITIALIZE BUFFERS & STATUS
                                        
173420  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
173424  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173432  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173436  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173440  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173444  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
173450  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173454  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
173460  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
173464  001412                          BEQ     5$
173466  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
173474  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173500  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173502  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173506  005302                          DEC     R2                      ;POINTER-1
173510  000763                          BR      4$
173512  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
173516  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173524  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173530  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173532  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173536  013702  172416                  MOV     @#IOBUF, R2             ;FETCH NEW POINTER
173542  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173546  022702  000000                  CMP     #0,     R2              ;POINTER=0
173552  001401                          BEQ     6$
173554  005204                          INC     R4                      ;REPORT ERROR
173556  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
173562  000207                          RTS     PC                      ;RETURN
173564  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR=RS
173572  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
173600  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
173606  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
173612  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
173616  022700  000000                  CMP     #0,     R0              ;STATUS=0 ?
173622  001401                          BEQ     7$
173624  005204                          INC     R4                      ;STATUS BAD REPORT
173626  022704  000000          7$:     CMP     #0,     R4              ;WERE THERE ERRORS
173632  001406                          BEQ     8$                      ;NO GOOD INIT
173634  005205                          INC     R5                      ;ERROR +1
173636  022705  000005                  CMP     #5,     R5              ;ERROR LIMIT YET
173642  001406                          BEQ     9$                      ;YES RETURN A 1(ERROR)
173644  000167  177220                  JMP     INIT                    ;NO RESTART VINIT
173650  012713  000001          8$:     MOV     #1,     (R3)            ;GOOD INIT.
173654  000167  000004                  JMP     SRTLDP
173660  012713  177777          9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
                                        .ENABLE LSB
```

```
                                        ;LOADER START
                                        
173664  005077  000764        SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
173670  005077  000762                CLR     @PKCNT                  ;PACKET COUNTER
173674  005277  000756                INC     @PKCNT                  ;START'S AT 1
173700  105037  172414                CLRB    @#CSR                   ;CLEAR CSR
173704  005737  172414        LIO:    TST     @#CSR                   ;REQUEST YET
173710  100375                      BPL     .-4                     ;NO LOOP ON LIO
173712  013700  174650                MOV     @#DATA,  R0             ;ADDRESS OF DATA
173716  005010                      CLR     (R0)                    ;CLEAR DATA
173720  012706  160500                MOV     #160500,  SP            ;RESET SP
173724  012737  010400  172416        MOV     #4352., @#OPREG         ;WCR : RS(0)
173732  012737  002400  172416        MOV     #1200., @#OPREG         ;RD
173740  013701  172416                MOV     @#IOBUF, R1
173744  042701  177400                BIC     #177400, R1             ;CLEAR UNUSED BITS
                                        
                                        ;BUFFER OVERFLOW
                                        
173750  132701  000004        B2$:    BITB    #BIT02, R1              ;OV-FL
173754  001410                      BEQ     B3$
173756  012702  010410                MOV     #4360., R2              ;COM=SEL INBUF 0
173762  012703  010440                MOV     #4384., R3              ;COM=RDBUFPNT
173766  004767  000144                JSR     PC, EMBF                ;EMPTY BUFFER
173772  004767  000530                JSR     PC, NAK                 ;SEND NAK
173776  132701  000010        B3$:    BITB    #BIT03, R1              ;OV-FL
174002  001410                      BEQ     B01$
174004  012702  010610                MOV     #4488., R2              ;COM=SEL INBUF 1
174010  012703  010640                MOV     #4512., R3              ;COM=RDBUFPNT
174014  004767  000116                JSR     PC, EMBF
174020  004767  000502                JSR     PC, NAK                 ;SEND NAK
                                        
                                        ;BUFFERS FULL
                                        
174024  132701  000001        B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
174030  001410                      BEQ     2$                      ;NO CK 1
174032  012702  010410        1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
174036  012703  010440                MOV     #4384., R3              ;COM=SEL INBUF0
174042  012704  000001                MOV     #BIT00, R4              ;WHICH CRC BIT
174046  004767  000026                JSR     PC,     PACK            ;GO FIND PACKET
174052  132701  000002        2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
174056  001712                      BEQ     LIO                     ;NO LOOP BACK
174060  012702  010610                MOV     #4488., R2              ;YES,COM=RDBUFADR
174064  012703  010640                MOV     #4512., R3              ;COM=SEL INBUF1
174070  012704  000002                MOV     #BIT01, R4              ;WHICH CRC BIT
174074  004767  000000                JSR     PC,     PACK            ;GO FIND PACKET
```

```
                                        ;UNLOAD BUFFERS
                                        
174100  005010          PACK:   CLR     (R0)                    ;CLEAR STATUS
174102  004767  000030          JSR     PC,     EMBF            ;EMPTY BUFFER
174106  005710                  TST     (R0)                    ;GOOD UNLOAD
174110  100003                  BPL     4$                      ;YES
174112  004767  000410  3$:     JSR     PC,     NAK             ;NO NAK TO REPLACE
174116  000207                  RTS     PC                      ;RETURN
174120  004767  000226  4$:     JSR     PC,     LOAD            ;LOAD + CONTROL PARSE
174124  005710                  TST     (R0)                    ;CHECK RESPONSE
174126  100771                  BMI     3$                      ;BAD NAK
174130  004767  000364          JSR     PC,     ACK             ;ACK IT AND CONTINUE
174134  000207                  RTS     PC
174136  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG         ;RS (CRC)
174144  012737  002400  172416          MOV     #1280., @#OPREG         ;RS CMD
174152  013705  172416                  MOV     @#IOBUF, R5             ;FETCH STATUS
174156  130405                          BITB    R4, R5                  ;CRC OK
174160  001002                          BNE     5$                      ;YES CONTINUE
174162  012710  177777                  MOV     #-1,    (R0)            ;FLAG BAD CRC
174166  012737  160000  172410  5$:     MOV     #160000, @#BAR          ;ADDRESS TO WRITE
174174  010237  172416                  MOV     R2,     @#OPREG         ;RDBUFADR
174200  012737  001400  172416          MOV     #768.,  @#OPREG         ;READ POINTER
174206  105737  172414                  TSTB    @#CSR                   ;GOOD RD
174212  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
174214  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
174220  013704  172416                  MOV     @#IOBUF, R4             ;READ POINTER
174224  042704  177400                  BIC     #177400, R4             ;CLEAR UNUSED BITS
174230  022704  000004                  CMP     #4,     R4              ;CONTROL PACKET
174234  001405                          BEQ     6$                      ;YES UNLOAD
174236  022704  000204                  CMP     #132.,  R4              ;MEMORY IMAGE UNLOAD
174242  001402                          BEQ     6$
174244  012710  177777                  MOV     #-1,    (R0)            ;BAD UNLOAD
174250  005404          6$:     NEG     R4                      ;2'S COMP
174252  010437  172412                  MOV     R4,  @#WCR              ;BYTE COUNT
174256  010337  172416                  MOV     R3,     @#OPREG         ;SEL BUFFER
174262  012737  001400  172416          MOV     #768.,  @#OPREG         ;FALSE RD
174270  105737  172414                  TSTB    @#CSR                   ;GOOD RD
174274  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
174276  105037  172414                  CLRB    @#CSR
174302  012737  021000  172416          MOV     #8704., @#OPREG         ;FIRE DMA RD
174310  000240                          NOP                             ;DELAY
174312  105737  172414                  TSTB    @#CSR                   ;GOOD DMA
174316  100402                          BMI     7$                      ;YES
174320  012710  177777                  MOV     #-1,  (R0)              ;BAD DMA XFER
174324  105037  172414          7$:     CLRB    @#CSR                   ;CLEAR CSR
174330  012737  004400  172416          MOV     #2304., @#OPREG         ;RESET POINTER TO 255
174336  105737  172414                  TSTB    @#CSR                   ;GOOD WD
174342  100375                          BPL     .-4                     ;NO RETRY
174344  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
174350  000207          8$:     RTS     PC
                                        .ENABLE LSB
```

```
                                        ;PACKET PARSE
                                        
174352  016702  000302          LOAD:   MOV     BUFFER, R2              ;BUFFER AREA
174356  122712  000000                  CMPB    #0, (R2)                ;MEMORY OR CONTROL
174362  001426                          BEQ     CMEQ                    ;CONTROL PACKET
174364  016703  000266                  MOV     PKCNT,  R3              ;PACKET COUNTER
174370  022713  000144                  CMP     #100.,  (R3)            ;LIMIT
174374  001002                          BNE     1$                      ;NO CONTINUE
174376  005013                          CLR     (R3)                    ;RESET POINTER
174400  005213                          INC     (R3)
174402  121312          1$:             CMPB    (R3),   (R2)            ;NEW PACKET
174404  001401                          BEQ     2$
174406  000207                          RTS     PC                      ;RETURN
174410  062702  000002          2$:     ADD     #2,     R2              ;OFFSET HEADER
174414  005213                          INC     (R3)                    ;COUNT + 1
174416  017703  000232                  MOV     @MEMPT, R3
174422  012704  000100                  MOV     #64.,   R4              ;BYTE COUNT
174426  012223          3$:             MOV     (R2)+, (R3)+            ;LOAD MEMORY
174430  077402                          SOB     R4,     3$              ;LOOP UNTIL ZERO
174432  010377  000216                  MOV     R3,     @MEMPT
174436  000207                          RTS     PC
174440  122762  000001  000001  CMEQ:   CMPB    #1,     1(R2)           ;IS IT A REPORT
174446  001004                          BNE     5$                      ;NO
174450  013703  174652                  MOV     @#STATUS, R3            ;LOCATION OF STATUS
174454  011310                          MOV     (R3),   (R0)            ;TRANSFER STATUS
174456  000207                          RTS     PC
174460  122702  000005  000001  5$:     CMPB    #5,     1(R2)           ;IS IT A START
174466  001005                          BNE     6$                      ;NO
174470  013700  000040                  MOV     @#40,   R0              ;FETCH START ADDRESS
174474  012706  001000                  MOV     #1000, SP               ;RESET USER STACK
174500  000110                          JMP     (R0)                    ;LEAVE PROM TO RAM
174502  122762  000006  000001  6$:     CMPB    #6,     1(R2)           ;WAS IT A LOAD COMMAND
174510  001002                          BNE     7$                      ;NO
174512  000167  177146                  JMP     SRTLDR                  ;YES GOTO LOAD START
174516  000207          7$:             RTS     PC
```

```
                                        ;ACK/NAK ROUTINE
                                        
174520  012705  174630          ACK:    MOV     #ACKPK,   R5            ;ADDRESS OF ACK
174524  000402                          BR      8$
174526  012705  174640          NAK:    MOV     #NAKPK,   R5            ;ADDRESS OF NAK
174532  012737  010540  172416  8$:     MOV     #4448.,  @#OPREG        ;SEL OUTBF0
174540  012504                          MOV     (R5)+,    R4            ;FETCH HEADER
174542  067704  000110                  ADD     @PKCNT,   R4            ;ADD PACKET COUNT
174546  010437  172416                  MOV     R4,      @#OPREG        ;WRITE IT
174552  105737  172414                  TSTB    @#CSR                   ;GOOD WD
174556  100375                          BPL     .-4                     ;NO LOOP
174560  012704  000003                  MOV     #3,       R4            ;BYTE COUNT
174564  012537  172416          9$:     MOV     (R5)+,   @#OPREG        ;WD
174570  105737  172414                  TSTB    @#CSR                   ;GOOD WD
174574  100375                          BPL     .-4                     ;NO LOOP
174576  077406                          SOB     R4,       9$            ;YES LOOP TIL 0
174600  012737  010420  172416          MOV     #4368.,  @#OPREG        ;MODSTAT
174606  012737  004421  172416          MOV     #4421,   @#OPREG        ;SET BITS
174614  105737  172414                  TSTB    @#CSR                   ;GOOD WD
174620  100375                          BPL     .-4                     ;LOOP UNTIL
174622  105037  172414                  CLRB    @#CSR
174626  000207                          RTS     PC

174630  004400                  ACKPK:  .WORD   4400
174632  004402                          .WORD   4402
174634  004403                          .WORD   4403
174636  004405                          .WORD   4405
174640  004400                  NAKPK:  .WORD   4400
174642  004403                          .WORD   4403
174644  004403                          .WORD   4403
174646  004405                          .WORD   4405
174650  160400                  DATA:   .WORD   160400
174652  160402                  STATUS: .WORD   160402
174654  160404                  MEMPT:  .WORD   160404
174656  160406                  PKCNT:  .WORD   160406
174660  160000                  BUFFER: .WORD   160000
174662  004404                  NODE:   .WORD   4404

        173000                          .END    START
```

```
PROM NODE 21   MACRO V03.01 27-MAR-79 18:54:23 PAGE 7-1
SYMBOL TABLE

ACK     174520          BIT06 = 000100          CMEQ    174440          LIO     173704          PACK    174100
ACKPK   174630          BIT07 = 000200          CSR   = 172414          LOAD    174352          PKCNT   174656
BAR   = 172410          BIT08 = 000400          DATA    174650          MEMPT   174654          SRTLDR  173664
BIT00 = 000001          BIT14 = 040000          EMBF    174136          NAK     174526          START   173000
BIT01 = 000002          BIT15 = 100000          INBUF0  173276          NAKPK   174640          STAT    173564
BIT02 = 000004          BUFFER  174660          INBUF1  173332          NODE    174662          STATUS  174652
BIT03 = 000010          B01$    174024          INIT    173070          OPREG = 172416          STRAM   173222
BIT04 = 000020          B2$     173750          INITLS  173014          OTBUF0  173366          WCR   = 172412
BIT05 = 000040          B3$     173770          IOBUF = 172416          OTBUF1  173402          ZEROBP  173420

. ABS.  174664    000
        000000    001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM21=DK:PROM21
```

```
                                        .TITLE PROM NODE 22
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000


        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1


173000  000005                  START:  RESET                   ;RESET BUS
173002  012700  000340                  MOV     #340,   R0      ;PRI=7
173006  106400                          MTPS    R0              ;CPU PRI
173010  012706  160500                  MOV     #160500, SP     ;RESET STACK

                                        ;INITIALIZE LINE SWITCHS

173014  012737  010420  172416  INITLS: MOV     #4368., @#OPREG ;MOD STAT
173022  012737  004414  172416          MOV     #4414,  @#OPREG ;SET SWITCHES
173030  105737  172414                  TSTB    @#CSR           ;GOOD WD
173034  100375                          BPL     .-4             ;LOOP UNTIL
173036  105037  172414                  CLRB    @#CSR
173042  012737  004540  172416          MOV     #4540,  @#OPREG ;RESET SWITCHS
173050  105737  172414                  TSTB    @#CSR           ;GOOD WD
173054  100375                          BPL     .-4             ;LOOP UNTIL
173056  105037  172414                  CLRB    @#CSR
173062  013705  174652                  MOV     @#STATUS, R3    ;LO CATION OF STATUS
173066  005005                          CLR     R5              ;CLEAR ERROR COUNT
```

```
                                        ;INITIALIZE LIU
                                        ;
173070  005004                  INIT:   CLR     R4                      ;CLEAR LOOP COUNT
173072  012737  010402  172416          MOV     #4354., @#OPREG         ;LDADR
173100  012737  004400  172416          MOV     #2304., @#OPREG         ;ADDRESS=0
173106  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173112  100375                          BPL     .-4                     ;NO RETRY
173114  105037  172414                  CLRB    @#CSR
173120  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
173126  012700  000400                  MOV     #256., R0               ;COUNTER
173132  012737  004407  172416  1$:     MOV     #2311., @#OPREG         ;WRITE A NULL
173140  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173144  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173146  105037  172414                  CLRB    @#CSR
173152  077011                          SOB     R0,     1$              ;LOOP UNTIL ZERO
173154  012700  000400                  MOV     #256.,  R0              ;COUNTER
173160  012737  001400  172416  2$:     MOV     #768.,  @#OPREG         ;RD
173166  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173172  100375                          BPL     .-4                     ;NO RETRY
173174  105037  172414                  CLRB    @#CSR
173200  013701  172416                  MOV     @#IOBUF, R1             ;FETCH DATA
173204  042701  177760                  BIC     #177760, R1             ;CLEAR BITS
173210  022701  000007                  CMP     #7,     R1              ;DATA=NULL
173214  001401                          BEQ     3$                      ;YES
173216  005204                          INC     R4                      ;NO REPORT IT
173220  077021                  3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO
173222  012737  010402  172416  STRAM:  MOV     #4354., @#OPREG         ;LDCTADR
173230  013737  174662  172416          MOV     @#NODE, @#OPREG         ;SEND ADDRESS
173236  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
173242  100375                          BPL     .-4                     ;LOOP UNTIL
173244  105037  172414                  CLRB    @#CSR
173250  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
173256  012737  004404  172416          MOV     #4404,  @#OPREG         ;LOAD A 4
173264  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
173270  100375                          BPL     .-4                     ;NO LOOP UNTIL
173272  105037  172414                  CLRB    @#CSR
173276  012701  010410          INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
173302  012700  010440                  MOV     #10440, R0              ;SEL INBUF0
173306  004767  000106                  JSR     PC,     ZEROBP          ;SET POINTER=0
173312  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
173320  105737  172414                  TSTB    @#CSR                   ;GOOD READ
173324  100375                          BPL     .-4                     ;NO RETRY
173326  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173332  012701  010610          INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
173336  012700  010640                  MOV     #10640, R0              ;SEL INBUF1 COMMAND
173342  004767  000052                  JSR     PC,     ZEROBP          ;POINTER=0
173346  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
173354  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173360  100375                          BPL     .-4                     ;NO RETRY
173362  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173366  012701  010510          OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
173372  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
173376  004767  000016                  JSR     PC,     ZEROBP
173402  012701  010710          OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
173406  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
173412  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
173416  000462                          BR      STAT                    ;GO CLEAR STATUS
```

```
                                        ;INITIALIZE BUFFERS & STATUS
                                        
173420  010137  172416          ZEROBF: MOV     R1,     @#OPREG         ;RDBUFADR
173424  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173432  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173436  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173440  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173444  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
173450  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173454  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
173460  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
173464  001412                          BEQ     5$
173466  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
173474  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173500  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173502  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173506  005302                          DEC     R2                      ;POINTER-1
173510  000763                          BR      4$
173512  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
173516  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173524  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173530  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173532  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173536  013702  172416                  MOV     @#IOBUF, R2             ;FETCH NEW POINTER
173542  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173546  022702  000000                  CMP     #0,     R2              ;POINTER=0
173552  001401                          BEQ     6$
173554  005204                          INC     R4                      ;REPORT ERROR
173556  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
173562  000207                          RTS     PC                      ;RETURN
173564  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR:RS
173572  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
173600  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
173606  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
173612  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
173616  022700  000000                  CMP     #0,     R0              ;STATUS=0 ?
173622  001401                          BEQ     7$
173624  005204                          INC     R4                      ;STATUS BAD REPORT
173626  022704  000000          7$:     CMP     #0,     R4              ;WERE THERE ERRORS
173632  001406                          BEQ     8$                      ;NO GOOD INIT
173634  005205                          INC     R5                      ;ERROR +1
173636  022705  000005                  CMP     #5,     R5              ;ERROR LIMIT YET
173642  001406                          BEQ     9$                      ;YES RETURN A 1(ERROR)
173644  000167  177220                  JMP     INIT                    ;NO RESTART VINIT
173650  012713  000001          8$:     MOV     #1,     (R3)            ;GOOD INIT.
173654  000167  000004                  JMP     SRTLDR
173660  012713  177777          9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
                                        .ENABLE LSB
```

```
 1                                          ;LOADER START
 2
 3 173664  005077  000764          SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
 4 173670  005077  000762                  CLR     @PKCNT                  ;PACKET COUNTER
 5 173674  005277  000756                  INC     @PKCNT                  ;START'S AT 1
 6 173700  105037  172414                  CLRB    @#CSR                   ;CLEAR CSR
 7 173704  005737  172414          LIO:    TST     @#CSR                   ;REQUEST YET
 8 173710  100375                          BPL     .-4                     ;NO LOOP ON LIO
 9 173712  013700  174650                  MOV     @#DATA,  R0             ;ADDRESS OF DATA
10 173716  005010                          CLR     (R0)                    ;CLEAR DATA
11 173720  012706  160500                  MOV     #160500,  SP            ;RESET SP
12 173724  012737  010400  172416          MOV     #4352., @#OPREG         ;WCR : RS(0)
13 173732  012737  002400  172416          MOV     #1280., @#OPREG         ;RD
14 173740  013701  172416                  MOV     @#IOBUF, R1
15 173744  042701  177400                  BIC     #177400, R1             ;CLEAR UNUSED BITS
16
17                                          ;BUFFER OVERFLOW
18
19 173750  132701  000004          B2$:    BITB    #BIT02, R1              ;OV-FL
20 173754  001410                          BEQ     B3$
21 173756  012702  010410                  MOV     #4360., R2              ;COM=SEL INBUF 0
22 173762  012703  010440                  MOV     #4384., R3              ;COM=RDBUFPNT
23 173766  004767  000144                  JSR     PC, EMBF                ;EMPTY BUFFER
24 173772  004767  000530                  JSR     PC, NAK                 ;SEND NAK
25 173776  132701  000010          B3$:    BITB    #BIT03, R1              ;OV-FL
26 174002  001410                          BEQ     B01$
27 174004  012702  010610                  MOV     #4488., R2              ;COM=SEL INBUF 1
28 174010  012703  010640                  MOV     #4512., R3              ;COM=RDBUFPNT
29 174014  004767  000116                  JSR     PC, EMBF
30 174020  004767  000502                  JSR     PC, NAK                 ;SEND NAK
31
32                                          ;BUFFERS FULL
33
34 174024  132701  000001          B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
35 174030  001410                          BEQ     2$                      ;NO CK 1
36 174032  012702  010410          1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
37 174036  012703  010440                  MOV     #4384., R3              ;COM=SEL INBUF0
38 174042  012704  000001                  MOV     #BIT00, R4              ;WHICH CRC BIT
39 174046  004767  000026                  JSR     PC,     PACK            ;GO FIND PACKET
40 174052  132701  000002          2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
41 174056  001712                          BEQ     LIO                     ;NO LOOP BACK
42 174060  012702  010610                  MOV     #4488., R2              ;YES,COM=RDBUFADR
43 174064  012703  010640                  MOV     #4512., R3              ;COM=SEL INBUF1
44 174070  012704  000002                  MOV     #BIT01, R4              ;WHICH CRC BIT
45 174074  004767  000000                  JSR     PC,     PACK            ;GO FIND PACKET
```

```
                                        ;UNLOAD BUFFERS
                                        
174100  005010                  PACK:   CLR     (R0)                    ;CLEAR STATUS
174102  004767  000030                  JSR     PC,     EMBF            ;EMPTY BUFFER
174106  005710                          TST     (R0)                    ;GOOD UNLOAD
174110  100003                          BPL     4$                      ;YES
174112  004767  000410          3$:     JSR     PC,     NAK             ;NO NAK TO REPLACE
174116  000207                          RTS     PC                      ;RETURN
174120  004767  000226          4$:     JSR     PC,     LOAD            ;LOAD + CONTROL PARSE
174124  005710                          TST     (R0)                    ;CHECK RESPONSE
174126  100771                          BMI     3$                      ;BAD NAK
174130  004767  000364                  JSR     PC,     ACK             ;ACK IT AND CONTINUE
174134  000207                          RTS     PC
174136  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG         ;RS (CRC)
174144  012737  002400  172416          MOV     #1280., @#OPREG         ;RS CMD
174152  013705  172416                  MOV     @#IOBUF, R5             ;FETCH STATUS
174156  130405                          BITB    R4, R5                  ;CRC OK
174160  001002                          BNE     5$                      ;YES CONTINUE
174162  012710  177777                  MOV     #-1,    (R0)            ;FLAG BAD CRC
174166  012737  160000  172410  5$:     MOV     #160000, @#BAR          ;ADDRESS TO WRITE
174174  010237  172416                  MOV     R2,     @#OPREG         ;RDBUFADR
174200  012737  001400  172416          MOV     #768.,  @#OPREG         ;READ POINTER
174206  105737  172414                  TSTB    @#CSR                   ;GOOD RD
174212  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
174214  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
174220  013704  172416                  MOV     @#IOBUF, R4             ;READ POINTER
174224  042704  177400                  BIC     #177400, R4             ;CLEAR UNUSED BITS
174230  022704  000004                  CMP     #4,     R4              ;CONTROL PACKET
174234  001405                          BEQ     6$                      ;YES UNLOAD
174236  022704  000204                  CMP     #132.,  R4              ;MEMORY IMAGE UNLOAD
174242  001402                          BEQ     6$
174244  012710  177777                  MOV     #-1,    (R0)            ;BAD UNLOAD
174250  005404                  6$:     NEG     R4                      ;2'S COMP
174252  010437  172412                  MOV     R4,  @#WCR              ;BYTE COUNT
174256  010337  172416                  MOV     R3,     @#OPREG         ;SEL BUFFER
174262  012737  001400  172416          MOV     #768.,  @#OPREG         ;FALSE RD
174270  105737  172414                  TSTB    @#CSR                   ;GOOD RD
174274  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
174276  105037  172414                  CLRB    @#CSR
174302  012737  021000  172416          MOV     #8704., @#OPREG         ;FIRE DMA RD
174310  000240                          NOP                             ;DELAY
174312  105737  172414                  TSTB    @#CSR                   ;GOOD DMA
174316  100402                          BMI     7$                      ;YES
174320  012710  177777                  MOV     #-1,  (R0)              ;BAD DMA XFER
174324  105037  172414          7$:     CLRB    @#CSR                   ;CLEAR CSR
174330  012737  004400  172416          MOV     #2304., @#OPREG         ;RESET POINTER TO 255
174336  105737  172414                  TSTB    @#CSR                   ;GOOD WD
174342  100375                          BPL     .-4                     ;NO RETRY
174344  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
174350  000207                  8$:     RTS     PC
                                        .ENABLE LSB
```

```
 1                                          ;PACKET PARSE
 2
 3 174352  016702  000302          LOAD:   MOV     BUFFER, R2          ;BUFFER AREA
 4 174356  122712  000000                  CMPB    #0, (R2)            ;MEMORY OR CONTROL
 5 174362  001426                          BEQ     CMEQ                ;CONTROL PACKET
 6 174364  016703  000266                  MOV     PKCNT,  R3          ;PACKET COUNTER
 7 174370  022713  000144                  CMP     #100.,  (R3)        ;LIMIT
 8 174374  001002                          BNE     1$                  ;NO CONTINUE
 9 174376  005013                          CLR     (R3)                ;RESET POINTER
10 174400  005213                          INC     (R3)
11 174402  121312                  1$:     CMPB    (R3),   (R2)        ;NEW PACKET
12 174404  001401                          BEQ     2$
13 174406  000207                          RTS     PC                  ;RETURN
14 174410  062702  000002          2$:     ADD     #2,     R2          ;OFFSET HEADER
15 174414  005213                          INC     (R3)                ;COUNT + 1
16 174416  017703  000232                  MOV     @MEMPT, R3
17 174422  012704  000100                  MOV     #64.,   R4          ;BYTE COUNT
18 174426  012223                  3$:     MOV     (R2)+, (R3)+        ;LOAD MEMORY
19 174430  077402                          SOB     R4,     3$          ;LOOP UNTIL ZERO
20 174432  010377  000216                  MOV     R3,     @MEMPT
21 174436  000207                          RTS     PC
22 174440  122762  000001  000001  CMEQ:   CMPB    #1,     1(R2)       ;IS IT A REPORT
23 174446  001004                          BNE     5$                  ;NO
24 174450  013703  174652                  MOV     @#STATUS, R3        ;LOCATION OF STATUS
25 174454  011310                          MOV     (R3),   (R0)        ;TRANSFER STATUS
26 174456  000207                          RTS     PC
27 174460  122762  000005  000001  5$:     CMPB    #5,     1(R2)       ;IS IT A START
28 174466  001005                          BNE     6$                  ;NO
29 174470  013700  000040                  MOV     @#40,   R0          ;FETCH START ADDRESS
30 174474  012706  001000                  MOV     #1000, SP           ;RESET USER STACK
31 174500  000110                          JMP     (R0)                ;LEAVE PROM TO RAM
32 174502  122762  000006  000001  6$:     CMPB    #6,     1(R2)       ;WAS IT A LOAD COMMAND
33 174510  001002                          BNE     7$                  ;NO
34 174512  000167  177146                  JMP     SRTIDR              ;YES GOTO LOAD START
35 174516  000207                  7$:     RTS     PC
36
```

```
 1                                                    ;ACK/NAK ROUTINE
 2
 3 174520  012705  174630          ACK:    MOV     #ACKPK,   R5              ;ADDRESS OF ACK
 4 174524  000402                          BR      8$
 5 174526  012705  174640          NAK:    MOV     #NAKPK,   R5              ;ADDRESS OF NAK
 6 174532  012737  010540  172416  8$:     MOV     #4448., @#OPREG           ;SEL OUTBFØ
 7 174540  012504                          MOV     (R5)+,  R4                ;FETCH HEADER
 8 174542  067704  000110                  ADD     @PKCNT, R4                ;ADD PACKET COUNT
 9 174546  010437  172416                  MOV     R4,     @#OPREG           ;WRITE IT
10 174552  105737  172414                  TSTB    @#CSR                     ;GOOD WD
11 174556  100375                          BPL     .-4                       ;NO LOOP
12 174560  012704  000003                  MOV     #3,     R4                ;BYTE COUNT
13 174564  012537  172416          9$:     MOV     (R5)+, @#OPREG            ;WD
14 174570  105737  172414                  TSTB    @#CSR                     ;GOOD WD
15 174574  100375                          BPL     .-4                       ;NO LOOP
16 174576  077406                          SOB     R4,     9$                ;YES LOOP TIL Ø
17 174600  012737  010420  172416          MOV     #4368., @#OPREG           ;MODSTAT
18 174606  012737  004421  172416          MOV     #4421,  @#OPREG           ;SET BITS
19 174614  105737  172414                  TSTB    @#CSR                     ;GOOD WD
20 174620  100375                          BPL     .-4                       ;LOOP UNTIL
21 174622  105037  172414                  CLRB    @#CSR
22 174626  000207                          RTS     PC
23
24
25 174630  004400                ACKPK:  .WORD   4400
26 174632  004402                        .WORD   4402
27 174634  004403                        .WORD   4403
28 174636  004405                        .WORD   4405
29 174640  004400                NAKPK:  .WORD   4400
30 174642  004403                        .WORD   4403
31 174644  004403                        .WORD   4403
32 174646  004405                        .WORD   4405
33 174650  160400                DATA:   .WORD   160400
34 174652  160402                STATUS: .WORD   160402
35 174654  160404                MEMPT:  .WORD   160404
36 174656  160406                PKCNT:  .WORD   160406
37 174660  160000                BUFFER: .WORD   160000
38 174662  004403                NODE:   .WORD   4403
39
40         173000                        .END    START
```

```
PROM NODE 22   MACRO V03.01 27-MAR-79 18:55:12 PAGE 7-1
SYMBOL TABLE

ACK     174520          BIT06 = 000100          CMEQ    174440          LIO     173704          PACK    174100
ACKPK   174630          BIT07 = 000200          CSR   = 172414          LOAD    174352          PKCNT   174656
BAR   = 172410          BIT08 = 000400          DATA    174650          MEMPT   174654          SRTLDR  173664
BIT00 = 000001          BIT14 = 040000          EMBF    174136          NAK     174526          START   173000
BIT01 = 000002          BIT15 = 100000          INBUF0  173276          NAKPK   174640          STAT    173564
BIT02 = 000004          BUFFER  174660          INBUF1  173332          NODE    174662          STATUS  174652
BIT03 = 000010          B01$    174024          INIT    173070          OPREG = 172416          STRAM   173222
BIT04 = 000020          B2$     173750          INITLS  173014          OTBUF0  173366          WCR   = 172412
BIT05 = 000040          B3$     173776          IOBUF = 172416          OTBUF1  173402          ZEROBP  173420

. ABS.  174664     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM22=DK:PROM22
```

```
                                        .TITLE PROM NODE 23
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000

        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1

173000  000005                  START:  RESET                   ;RESET BUS
173002  012700  000340                  MOV     #340,   R0      ;PRI=7
173006  106400                          MTPS    R0              ;CPU PRI
173010  012706  160500                  MOV     #160500, SP     ;RESET STACK

                                        ;INITIALIZE LINE SWITCHS

173014  012737  010420  172416  INITLS: MOV     #4368., @#OPREG ;MOD STAT
173022  012737  004414  172416          MOV     #4414,  @#OPREG ;SET S ITCHES
173030  105737  172414                  TSTB    @#CSR           ;GOOD WD
173034  100375                          BPL     .-4             ;LOOP UNTIL
173036  105037  172414                  CLRB    @#CSR
173042  012737  004540  172416          MOV     #4540,  @#OPREG ;RESET SWITCHS
173050  105737  172414                  TSTB    @#CSR           ;GOOD WD
173054  100375                          BPL     .-4             ;LOOP UNTIL
173056  105037  172414                  CLRB    @#CSR
173062  013703  174652                  MOV     @#STATUS, R3    ;LO CATION OF STATUS
173066  005005                          CLR     R5              ;CLEAR ERROR COUNT
```

```
 1                                                  ;INITIALIZE LIU
 2
 3 173070  005004                        INIT:     CLR     R4                    ;CLEAR LOOP COUNT
 4 173072  012737  010402  172416                  MOV     #4354., @#OPREG       ;LDADR
 5 173100  012737  004400  172416                  MOV     #2304., @#OPREG       ;ADDRESS=0
 6 173106  105737  172414                          TSTB    @#CSR                 ;GOOD WD
 7 173112  100375                                  BPL     .-4                   ;NO RETRY
 8 173114  105037  172414                          CLRB    @#CSR
 9 173120  012737  010401  172416                  MOV     #4353., @#OPREG       ;SEL ACRAM
10 173126  012700  000400                          MOV     #256., R0             ;COUNTER
11 173132  012737  004407  172416          1$:     MOV     #2311., @#OPREG       ;WRITE A NULL
12 173140  105737  172414                          TSTB    @#CSR                 ;GOOD WD
13 173144  100375                                  BPL     .-4                   ;NO LOOP UNTIL READY
14 173146  105037  172414                          CLRB    @#CSR
15 173152  077011                                  SOB     R0,     1$            ;LOOP UNTIL ZERO
16 173154  012700  000400                          MOV     #256.,  R0            ;COUNTER
17 173160  012737  001400  172416          2$:     MOV     #768.,  @#OPREG       ;RD
18 173166  105737  172414                          TSTB    @#CSR                 ;GOOD RD
19 173172  100375                                  BPL     .-4                   ;NO RETRY
20 173174  105037  172414                          CLRB    @#CSR
21 173200  013701  172416                          MOV     @#IOBUF, R1           ;FETCH DATA
22 173204  042701  177760                          BIC     #177760, R1           ;CLEAR BITS
23 173210  022701  000007                          CMP     #7,     R1            ;DATA=NULL
24 173214  001401                                  BEQ     3$                    ;YES
25 173216  005204                                  INC     R4                    ;NO REPORT IT
26 173220  077021                          3$:     SOB     R0,     2$            ;LOOP UNTIL ZERO
27 173222  012737  010402  172416          STRAM:  MOV     #4354., @#OPREG       ;LDCTADR
28 173230  013737  174662  172416                  MOV     @#NODE, @#OPREG       ;SEND ADDRESS
29 173236  105737  172414                          TSTB    @#CSR                 ;GOOD WRITE
30 173242  100375                                  BPL     .-4                   ;LOOP UNTIL
31 173244  105037  172414                          CLRB    @#CSR
32 173250  012737  010401  172416                  MOV     #4353., @#OPREG       ;SEL ACRAM
33 173256  012737  004404  172416                  MOV     #4404,  @#OPREG       ;LOAD A 4
34 173264  105737  172414                          TSTB    @#CSR                 ;GOOD WRITE
35 173270  100375                                  BPL     .-4                   ;NO LOOP UNTIL
36 173272  105037  172414                          CLRB    @#CSR
37 173276  012701  010410                  INBUF0: MOV     #10410, R1            ;RDBUFADR CMD
38 173302  012700  010440                          MOV     #10440, R0            ;SEL INBUF0
39 173306  004767  000106                          JSR     PC,     ZEROBP        ;SET POINTER=0
40 173312  012737  001400  172416                  MOV     #1400,  @#OPREG       ;FALSE READ DATA
41 173320  105737  172414                          TSTB    @#CSR                 ;GOOD READ
42 173324  100375                                  BPL     .-4                   ;NO RETRY
43 173326  105037  172414                          CLRB    @#CSR                 ;CLEAR DONE BIT
44 173332  012701  010610                  INBUF1: MOV     #10610, R1            ;RDBUFADR COMMAND
45 173336  012700  010640                          MOV     #10640, R0            ;SEL INBUF1 COMMAND
46 173342  004767  000052                          JSR     PC,     ZEROBP        ;POINTER=0
47 173346  012737  001400  172416                  MOV     #1400,  @#OPREG       ;FALSE READ DATA
48 173354  105737  172414                          TSTB    @#CSR                 ;GOOD RD
49 173360  100375                                  BPL     .-4                   ;NO RETRY
50 173362  105037  172414                          CLRB    @#CSR                 ;CLEAR DONE BIT
51 173366  012701  010510                  OTBUF0: MOV     #10510, R1            ;RDBUFADR COMMAND
52 173372  012700  010540                          MOV     #10540, R0            ;SEL OUTBUF0 COMMAND
53 173376  004767  000016                          JSR     PC,     ZEROBP
54 173402  012701  010710                  OTBUF1: MOV     #10710, R1            ;RDBUFADR COMMAND
55 173406  012700  010740                          MOV     #10740, R0            ;SEL OUTBUF0 COMMAND
56 173412  004767  000002                          JSR     PC,     ZEROBP        ;ZERO BUFFER POINTER
57 173416  000462                                  BR      STAT                  ;GO CLEAR STATUS
```

```
                                        ;INITIALIZE BUFFERS & STATUS

173420  010137  172416        ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
173424  012737  001400  172416        MOV     #1400,  @#OPREG         ;RD
173432  105737  172414                TSTB    @#CSR                   ;GOOD RD
173436  100375                        BPL     .-4                     ;NO LOOP UNTIL READY
173440  105037  172414                CLRB    @#CSR                   ;CLEAR DONE BIT
173444  013702  172416                MOV     @#IOBUF, R2             ;FETCH POINTER
173450  042702  177400                BIC     #177400, R2             ;CLEAR MST BYTE
173454  010037  172416                MOV     R0,     @#OPREG         ;SEL BUFFER
173460  022702  000000        4$:     CMP     #0,     R2              ;POINTER=0
173464  001412                        BEQ     5$
173466  012737  001400  172416        MOV     #1400,  @#OPREG         ;NO FALSE RD
173474  105737  172414                TSTB    @#CSR                   ;GOOD RD
173500  100375                        BPL     .-4                     ;NO LOOP UNTIL READY
173502  105037  172414                CLRB    @#CSR                   ;CLEAR DONE BIT
173506  005302                        DEC     R2                      ;POINTER-1
173510  000763                        BR      4$
173512  010137  172416        5$:     MOV     R1,     @#OPREG         ;RDBUFADR
173516  012737  001400  172416        MOV     #1400,  @#OPREG         ;RD
173524  105737  172414                TSTB    @#CSR                   ;GOOD RD
173530  100375                        BPL     .-4                     ;NO LOOP UNTIL READY
173532  105037  172414                CLRB    @#CSR                   ;CLEAR DONE BIT
173536  013702  172416                MOV     @#IOBUF, R2             ;FETCH NEW POINTER
173542  042702  177400                BIC     #177400, R2             ;CLEAR MST BYTE
173546  022702  000000                CMP     #0,     R2              ;POINTER=0
173552  001401                        BEQ     6$
173554  005204                        INC     R4                      ;REPORT ERROR
173556  010037  172416        6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
173562  000207                        RTS     PC                      ;RETURN
173564  012737  010400  172416  STAT: MOV     #4352., @#OPREG         ;WCR:RS
173572  012737  002400  172416        MOV     #1280., @#OPREG         ;RS(FALSE)
173600  012737  002400  172416        MOV     #1280., @#OPREG         ;RS
173606  013700  172416                MOV     @#OPREG, R0             ;FETCH STATUS
173612  042700  177400                BIC     #177400, R0             ;CLEAR MST BYTE
173616  022700  000000                CMP     #0,     R0              ;STATUS=0 ?
173622  001401                        BEQ     7$
173624  005204                        INC     R4                      ;STATUS BAD REPORT
173626  022704  000000        7$:     CMP     #0,     R4              ;WERE THERE ERRORS
173632  001406                        BEQ     8$                      ;NO GOOD INIT
173634  005205                        INC     R5                      ;ERROR +1
173636  022705  000005                CMP     #5,     R5              ;ERROR LIMIT YET
173642  001406                        BEQ     9$                      ;YES RETURN A 1(ERROR)
173644  000167  177220                JMP     INIT                    ;NO RESTART VINIT
173650  012713  000001        8$:     MOV     #1,     (R3)            ;GOOD INIT.
173654  000167  000004                JMP     SRTLDR
173660  012713  177777        9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
                                .ENABLE LSB
```

```
 1                                              ;LOADER START
 2
 3 173664  005077  000764          SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
 4 173670  005077  000762                  CLR     @PKCNT                  ;PACKET COUNTER
 5 173674  005277  000756                  INC     @PKCNT                  ;START'S AT 1
 6 173700  105037  172414                  CLRB    @#CSR                   ;CLEAR CSR
 7 173704  005737  172414          LIO:    TST     @#CSR                   ;REQUEST YET
 8 173710  100375                          BPL     .-4                     ;NO LOOP ON LIO
 9 173712  013700  174650                  MOV     @#DATA,  R0             ;ADDRESS OF DATA
10 173716  005010                          CLR     (R0)                    ;CLEAR DATA
11 173720  012706  160500                  MOV     #160500,  SP            ;RESET SP
12 173724  012737  010400  172416          MOV     #4352., @#OPREG         ;WCR : RS(0)
13 173732  012737  002400  172416          MOV     #1280., @#OPREG         ;RD
14 173740  013701  172416                  MOV     @#IOBUF, R1
15 173744  042701  177400                  BIC     #177400, R1             ;CLEAR UNUSED BITS
16
17                                              ;BUFFER OVERFLOW
18
19 173750  132701  000004          B2$:    BITB    #BIT02, R1              ;OV-FL
20 173754  001410                          BEQ     B3$
21 173756  012702  010410                  MOV     #4360., R2              ;COM=SEL INBUF 0
22 173762  012703  010440                  MOV     #4384., R3              ;COM=RDBUFPNT
23 173766  004767  000144                  JSR     PC, EMBF                ;EMPTY BUFFER
24 173772  004767  000530                  JSR     PC, NAK                 ;SEND NAK
25 173776  132701  000010          B3$:    BITB    #BIT03, R1              ;OV-FL
26 174002  001410                          BEQ     B01$
27 174004  012702  010610                  MOV     #4488., R2              ;COM=SEL INBUF 1
28 174010  012703  010640                  MOV     #4512., R3              ;COM=RDBUFPNT
29 174014  004767  000116                  JSR     PC, EMBF
30 174020  004767  000502                  JSR     PC, NAK                 ;SEND NAK
31
32                                              ;BUFFERS FULL
33
34 174024  132701  000001          B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
35 174030  001410                          BEQ     2$                      ;NO CK 1
36 174032  012702  010410          1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
37 174036  012703  010440                  MOV     #4384., R3              ;COM=SEL INBUF0
38 174042  012704  000001                  MOV     #BIT00, R4              ;WHICH CRC BIT
39 174046  004767  000026                  JSR     PC,     PACK            ;GO FIND PACKET
40 174052  132701  000002          2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
41 174056  001712                          BEQ     LIO                     ;NO LOOP BACK
42 174060  012702  010610                  MOV     #4488., R2              ;YES,COM=RDBUFADR
43 174064  012703  010640                  MOV     #4512., R3              ;COM=SEL INBUF1
44 174070  012704  000002                  MOV     #BIT01, R4              ;WHICH CRC BIT
45 174074  004767  000000                  JSR     PC,     PACK            ;GO FIND PACKET
```

```
                                        ;UNLOAD BUFFERS
                                        
174100  005010                  PACK:   CLR     (R0)            ;CLEAR STATUS
174102  004767  000030                  JSR     PC,     EMBF    ;EMPTY BUFFER
174106  005710                          TST     (R0)            ;GOOD UNLOAD
174110  100003                          BPL     4$              ;YES
174112  004767  000410          3$:     JSR     PC,     NAK     ;NO NAK TO REPLACE
174116  000207                          RTS     PC              ;RETURN
174120  004767  000226          4$:     JSR     PC,     LOAD    ;LOAD + CONTROL PARSE
174124  005710                          TST     (R0)            ;CHECK RESPONSE
174126  100771                          BMI     3$              ;BAD NAK
174130  004767  000364                  JSR     PC,     ACK     ;ACK IT AND CONTINUE
174134  000207                          RTS     PC
174136  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG ;RS (CRC)
174144  012737  002400  172416          MOV     #1280., @#OPREG ;RS CMD
174152  013705  172416                  MOV     @#IOBUF, R5     ;FETCH STATUS
174156  130405                          BITB    R4, R5          ;CRC OK
174160  001002                          BNE     5$              ;YES CONTINUE
174162  012710  177777                  MOV     #-1,    (R0)    ;FLAG BAD CRC
174166  012737  160000  172410  5$:     MOV     #160000, @#BAR  ;ADDRESS TO WRITE
174174  010237  172416                  MOV     R2,     @#OPREG ;RDBUFADR
174200  012737  001400  172416          MOV     #768.,  @#OPREG ;READ POINTER
174206  105737  172414                  TSTB    @#CSR           ;GOOD RD
174212  100375                          BPL     .-4             ;NO LOOP UNTIL READY
174214  105037  172414                  CLRB    @#CSR           ;CLEAR DONE BIT
174220  013704  172416                  MOV     @#IOBUF, R4     ;READ POINTER
174224  042704  177400                  BIC     #177400, R4     ;CLEAR UNUSED BITS
174230  022704  000004                  CMP     #4,     R4      ;CONTROL PACKET
174234  001405                          BEQ     6$              ;YES UNLOAD
174236  022704  000204                  CMP     #132.,  R4      ;MEMORY IMAGE UNLOAD
174242  001402                          BEQ     6$
174244  012710  177777                  MOV     #-1,    (R0)    ;BAD UNLOAD
174250  005404                  6$:     NEG     R4              ;2'S COMP
174252  010437  172412                  MOV     R4,  @#WCR      ;BYTE COUNT
174256  010337  172416                  MOV     R3,     @#OPREG ;SEL BUFFER
174262  012737  001400  172416          MOV     #768.,  @#OPREG ;FALSE RD
174270  105737  172414                  TSTB    @#CSR           ;GOOD RD
174274  100375                          BPL     .-4             ;NO LOOP UNTIL READY
174276  105037  172414                  CLRB    @#CSR
174302  012737  021000  172416          MOV     #8704., @#OPREG ;FIRE BMA RD
174310  000240                          NOP                     ;DELAY
174312  105737  172414                  TSTB    @#CSR           ;GOOD BMA
174316  100402                          BMI     7$              ;YES
174320  012710  177777                  MOV     #-1,  (R0)      ;BAD DMA XFER
174324  105037  172414          7$:     CLRB    @#CSR           ;CLEAR CSR
174330  012737  004400  172416          MOV     #2304., @#OPREG ;RESET POINTER TO 255
174336  105737  172414                  TSTB    @#CSR           ;GOOD WD
174342  100375                          BPL     .-4             ;NO RETRY
174344  105037  172414                  CLRB    @#CSR           ;CLEAR DONE BIT
174350  000207                  8$:     RTS     PC
                                        .ENABLE LSB
```

```
 1                                                  ;PACKET PARSE
 2
 3 174352  016702  000302          LOAD:   MOV     BUFFER, R2          ;BUFFER AREA
 4 174356  122712  000000                  CMPB    #0, (R2)            ;MEMORY OR CONTROL
 5 174362  001426                          BEQ     CMEQ                ;CONTROL PACKET
 6 174364  016703  000266                  MOV     PKCNT,  R3          ;PACKET COUNTER
 7 174370  022713  000144                  CMP     #100.,  (R3)        ;LIMIT
 8 174374  001002                          BNE     1$                  ;NO CONTINUE
 9 174376  005013                          CLR     (R3)                ;RESET POINTER
10 174400  005213                          INC     (R3)
11 174402  121312                 1$:     CMPB    (R3),   (R2)        ;NEW PACKET
12 174404  001401                          BEQ     2$
13 174406  000207                          RTS     PC                  ;RETURN
14 174410  062702  000002          2$:     ADD     #2,     R2          ;OFFSET HEADER
15 174414  005213                          INC     (R3)                ;COUNT + 1
16 174416  017703  000232                  MOV     @MEMPT, R3
17 174422  012704  000100                  MOV     #64.,   R4          ;BYTE COUNT
18 174426  012223                 3$:     MOV     (R2)+, (R3)+        ;LOAD MEMORY
19 174430  077402                          SOB     R4,     3$          ;LOOP UNTIL ZERO
20 174432  010377  000216                  MOV     R3,     @MEMPT
21 174436  000207                          RTS     PC
22 174440  122762  000001  000001  CMEQ:   CMPB    #1,     1(R2)       ;IS IT A REPORT
23 174446  001004                          BNE     5$                  ;NO
24 174450  013703  174652                  MOV     @#STATUS, R3        ;LOCATION OF STATUS
25 174454  011310                          MOV     (R3),   (R0)        ;TRANSFER STATUS
26 174456  000207                          RTS     PC
27 174460  122762  000005  000001  5$:     CMPB    #5,     1(R2)       ;IS IT A START
28 174466  001005                          BNE     6$                  ;NO
29 174470  013700  000040                  MOV     @#40,   R0          ;FETCH START ADDRESS
30 174474  012706  001000                  MOV     #1000, SP           ;RESET USER STACK
31 174500  000110                          JMP     (R0)                ;LEAVE PROM TO RAM
32 174502  122762  000006  000001  6$:     CMPB    #6,     1(R2)       ;WAS IT A LOAD COMMAND
33 174510  001002                          BNE     7$                  ;NO
34 174512  000167  177146                  JMP     SRTLDR              ;YES GOTO LOAD START
35 174516  000207                 7$:     RTS     PC
36
```

```
                                        ;ACK/NAK ROUTINE

174520  012705  174630        ACK:    MOV     #ACKPK,   R5            ;ADDRESS OF ACK
174524  000402                        BR      8$
174526  012705  174640        NAK:    MOV     #NAKPK,   R5            ;ADDRESS OF NAK
174532  012737  010540  172416 8$:     MOV     #4448., @#OPREG         ;SEL OUTBF0
174540  012504                        MOV     (R5)+,  R4              ;FETCH HEADER
174542  067704  000110                ADD     @PKCNT, R4              ;ADD PACKET COUNT
174546  010437  172416                MOV     R4,     @#OPREG         ;WRITE IT
174552  105737  172414                TSTB    @#CSR                   ;GOOD WD
174556  100375                        BPL     .-4                     ;NO LOOP
174560  012704  000003                MOV     #3,     R4              ;BYTE COUNT
174564  012537  172416        9$:     MOV     (R5)+, @#OPREG          ;WD
174570  105737  172414                TSTB    @#CSR                   ;GOOD WD
174574  100375                        BPL     .-4                     ;NO LOOP
174576  077406                        SOB     R4,     9$              ;YES LOOP TIL 0
174600  012737  010420  172416        MOV     #4368., @#OPREG         ;MODSTAT
174606  012737  004421  172416        MOV     #4421,  @#OPREG         ;SET BITS
174614  105737  172414                TSTB    @#CSR                   ;GOOD WD
174620  100375                        BPL     .-4                     ;LOOP UNTIL
174622  105037  172414                CLRB    @#CSR
174626  000207                        RTS     PC

174630  004400        ACKPK:  .WORD   4400
174632  004402                .WORD   4402
174634  004403                .WORD   4403
174636  004405                .WORD   4405
174640  004400        NAKPK:  .WORD   4400
174642  004403                .WORD   4403
174644  004403                .WORD   4403
174646  004405                .WORD   4405
174650  160400        DATA:   .WORD   160400
174652  160402        STATUS: .WORD   160402
174654  160404        MEMPT:  .WORD   160404
174656  160406        PKCNT:  .WORD   160406
174660  160000        BUFFER: .WORD   160000
174662  004406        NODE:   .WORD   4406

              173000                .END    START
```

```
PRCM NODE 23   MACRO V03.01 27-MAR-79 18:56:02 PAGE 7-1
SYMBOL TABLE

ACK     174520          BIT06 = 000100          CMEQ    174440          LIO     173704          PACK    174100
ACKPK   174630          BIT07 = 000200          CSR   = 172414          LOAD    174352          PKCNT   174656
BAR   = 172410          BIT08 = 000400          DATA    174650          MEMPT   174654          SRTLDR  173664
BIT00 = 000001          BIT14 = 040000          EMBF    174136          NAK     174526          START   173000
BIT01 = 000002          BIT15 = 100000          INBUF0  173276          NAKPK   174640          STAT    173564
BIT02 = 000004          BUFFER  174660          INBUF1  173332          NODE    174662          STATUS  174652
BIT03 = 000010          B01$    174024          INIT    173070          OPREG = 172416          STRAM   173222
BIT04 = 000020          B2$     173750          INITLS  173014          OTBUF0  173366          WCR   = 172412
BIT05 = 000040          B3$     173776          IOBUF = 172416          OTBUF1  173402          ZEROBP  173420

. ABS.  174664      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM23=DK:PROM23
```

```
                                        .TITLE PROM NODE 25
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000

                                BAR=    172410
                                WCR=    172412
                                CSR=    172414
                                IOBUF=  172416
                                OPREG=  172416
                                XCSR=   177564
                                XBUF=   177566
                                BIT15=  100000
                                BIT14=  40000
                                BIT08=  400
                                BIT07=  200
                                BIT06=  100
                                BIT05=  40
                                BIT04=  20
                                BIT03=  10
                                BIT02=  4
                                BIT01=  2
                                BIT00=  1

173000  000005          START:  RESET                   ;RESET BUS
173002  012700  000340        MOV     #340,   R0      ;PRI=7
173006  106400                MTPS    R0              ;CPU PRI
173010  012706  160500        MOV     #160500, SP     ;RESET STACK

                                ;INITIALIZE LINE SWITCHS

173014  012737  010420  172416 INITLS: MOV     #4368., @#OPREG ;MOD STAT
173022  012737  004414  172416        MOV     #4414,  @#OPREG ;SET S ITCHES
173030  105737  172414                TSTB    @#CSR           ;GOOD WD
173034  100375                        BPL     .-4             ;LOOP UNTIL
173036  105037  172414                CLRB    @#CSR
173042  012737  004540  172416        MOV     #4540,  @#OPREG ;RESET SWITCHS
173050  105737  172414                TSTB    @#CSR           ;GOOD WD
173054  100375                        BPL     .-4             ;LOOP UNTIL
173056  105037  172414                CLRB    @#CSR
173062  013703  174712                MOV     @#STATUS, R3    ;LO CATION OF STATUS
173066  005005                        CLR     R5              ;CLEAR ERROR COUNT
```

```
 1                                                ;INITIALIZE LIU
 2
 3 173070  005004                        INIT:   CLR     R4                      ;CLEAR LOOP COUNT
 4 173072  012737  010402  172416                MOV     #4354., @#OPREG         ;LDADR
 5 173100  012737  004400  172416                MOV     #2304., @#OPREG         ;ADDRESS=0
 6 173106  105737  172414                        TSTB    @#CSR                   ;GOOD WD
 7 173112  100375                                BPL     .-4                     ;NO RETRY
 8 173114  105037  172414                        CLRB    @#CSR
 9 173120  012737  010401  172416                MOV     #4353., @#OPREG         ;SEL ACRAM
10 173126  012700  000400                        MOV     #256., R0               ;COUNTER
11 173132  012737  004407  172416  1$:           MOV     #2311., @#OPREG         ;WRITE A NULL
12 173140  105737  172414                        TSTB    @#CSR                   ;GOOD WD
13 173144  100375                                BPL     .-4                     ;NO LOOP UNTIL READY
14 173146  105037  172414                        CLRB    @#CSR
15 173152  077011                                SOB     R0,     1$              ;LOOP UNTIL ZERO
16 173154  012700  000400                        MOV     #256.,  R0              ;COUNTER
17 173160  012737  001400  172416  2$:           MOV     #768.,  @#OPREG         ;RD
18 173166  105737  172414                        TSTB    @#CSR                   ;GOOD RD
19 173172  100375                                BPL     .-4                     ;NO RETRY
20 173174  105037  172414                        CLRB    @#CSR
21 173200  013701  172416                        MOV     @#IOBUF, R1             ;FETCH DATA
22 173204  042701  177760                        BIC     #177760, R1             ;CLEAR BITS
23 173210  022701  000007                        CMP     #7,      R1             ;DATA=NULL
24 173214  001401                                BEQ     3$                      ;YES
25 173216  005204                                INC     R4                      ;NO REPORT IT
26 173220  077021                        3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO
27 173222  012737  010402  172416  STRAM:  MOV     #4354., @#OPREG         ;LDCTADR
28 173230  013737  174722  172416                MOV     @#NODE, @#OPREG         ;SEND ADDRESS
29 173236  105737  172414                        TSTB    @#CSR                   ;GOOD WRITE
30 173242  100375                                BPL     .-4                     ;LOOP UNTIL
31 173244  105037  172414                        CLRB    @#CSR
32 173250  012737  010401  172416                MOV     #4353., @#OPREG         ;SEL ACRAM
33 173256  012737  004404  172416                MOV     #4404,  @#OPREG         ;LOAD A 4
34 173264  105737  172414                        TSTB    @#CSR                   ;GOOD WRITE
35 173270  100375                                BPL     .-4                     ;NO LOOP UNTIL
36 173272  105037  172414                        CLRB    @#CSR
37 173276  012701  010410                INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
38 173302  012700  010440                        MOV     #10440, R0              ;SEL INBUF0
39 173306  004767  000106                        JSR     PC,     ZEROBP          ;SET POINTER=0
40 173312  012737  001400  172416                MOV     #1400,  @#OPREG         ;FALSE READ DATA
41 173320  105737  172414                        TSTB    @#CSR                   ;GOOD READ
42 173324  100375                                BPL     .-4                     ;NO RETRY
43 173326  105037  172414                        CLRB    @#CSR                   ;CLEAR DONE BIT
44 173332  012701  010610                INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
45 173336  012700  010640                        MOV     #10640, R0              ;SEL INBUF1 COMMAND
46 173342  004767  000052                        JSR     PC,     ZEROBP          ;POINTER=0
47 173346  012737  001400  172416                MOV     #1400,  @#OPREG         ;FALSE READ DATA
48 173354  105737  172414                        TSTB    @#CSR                   ;GOOD RD
49 173360  100375                                BPL     .-4                     ;NO RETRY
50 173362  105037  172414                        CLRB    @#CSR                   ;CLEAR DONE BIT
51 173366  012701  010510                OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
52 173372  012700  010540                        MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
53 173376  004767  000016                        JSR     PC,     ZEROBP
54 173402  012701  010710                OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
55 173406  012700  010740                        MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
56 173412  004767  000002                        JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
57 173416  000462                                BR      STAT                    ;GO CLEAR STATUS
```

```
 1                                          ;INITIALIZE BUFFERS & STATUS
 2
 3 173420  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
 4 173424  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
 5 173432  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 6 173436  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
 7 173440  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 8 173444  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
 9 173450  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
10 173454  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
11 173460  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
12 173464  001412                          BEQ     5$
13 173466  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
14 173474  105737  172414                  TSTB    @#CSR                   ;GOOD RD
15 173500  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
16 173502  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
17 173506  005302                          DEC     R2                      ;POINTER-1
18 173510  000763                          BR      4$
19 173512  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
20 173516  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
21 173524  105737  172414                  TSTB    @#CSR                   ;GOOD RD
22 173530  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
23 173532  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
24 173536  013702  172416                  MOV     @#IOBUF, R2             ;FETCH NEW POINTER
25 173542  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
26 173546  022702  000000                  CMP     #0,     R2              ;POINTER=0
27 173552  001401                          BEQ     6$
28 173554  005204                          INC     R4                      ;REPORT ERROR
29 173556  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
30 173562  000207                          RTS     PC                      ;RETURN
31 173564  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR:RS
32 173572  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
33 173600  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
34 173606  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
35 173612  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
36 173616  022700  000000                  CMP     #0,     R0              ;STATUS=0 ?
37 173622  001401                          BEQ     7$
38 173624  005204                          INC     R4                      ;STATUS BAD REPORT
39 173626  022704  000000          7$:     CMP     #0,     R4              ;WERE THERE ERRORS
40 173632  001406                          BEQ     8$                      ;NO GOOD INIT
41 173634  005205                          INC     R5                      ;ERROR +1
42 173636  022705  000005                  CMP     #5,     R5              ;ERROR LIMIT YET
43 173642  001406                          BEQ     9$                      ;YES RETURN A 1(ERROR)
44 173644  000167  177220                  JMP     INIT                    ;NO RESTART VINIT
45 173650  012713  000001          8$:     MOV     #1,     (R3)            ;GOOD INIT.
46 173654  000167  000004                  JMP     SRTLDH
47 173660  012713  177777          9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
48                                          .ENABLE LSB
```

```
 1                                          ;LOADER START
 2
 3 173664  005077  001024          SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
 4 173670  005077  001022                  CLR     @PKCNT                  ;PACKET COUNTER
 5 173674  005277  001016                  INC     @PKCNT                  ;START'S AT 1
 6 173700  105037  172414                  CLRB    @#CSR                   ;CLEAR CSR
 7 173704  005737  172414          LIO:    TST     @#CSR                   ;REQUEST YET
 8 173710  100375                  BPL     .-4                             ;NO LOOP ON LIO
 9 173712  013700  174710                  MOV     @#DATA,  R0             ;ADDRESS OF DATA
10 173716  005010                  CLR     (R0)                            ;CLEAR DATA
11 173720  012706  160500                  MOV     #160500,  SP            ;RESET SP
12 173724  012737  010400  172416          MOV     #4352., @#OPREG         ;WCR : RS(0)
13 173732  012737  002400  172416          MOV     #1280., @#OPREG         ;RD
14 173740  013701  172416                  MOV     @#IOBUF, R1
15 173744  042701  177400                  BIC     #177400, R1             ;CLEAR UNUSED BITS
16
17                                          ;BUFFER OVERFLOW
18
19 173750  132701  000004          B2$:    BITB    #BIT02, R1              ;OV-FL
20 173754  001410                  BEQ     B3$
21 173756  012702  010410                  MOV     #4360., R2              ;COM=SEL INBUF 0
22 173762  012703  010440                  MOV     #4384., R3              ;COM=RDBUFPNT
23 173766  004767  000144                  JSR     PC, EMBF                ;EMPTY BUFFER
24 173772  004767  000570                  JSR     PC, NAK                 ;SEND NAK
25 173776  132701  000010          B3$:    BITB    #BIT03, R1              ;OV-FL
26 174002  001410                  BEQ     B01$
27 174004  012702  010610                  MOV     #4488., R2              ;COM=SEL INBUF 1
28 174010  012703  010640                  MOV     #4512., R3              ;COM=RDBUFPNT
29 174014  004767  000116                  JSR     PC, EMBF
30 174020  004767  000542                  JSR     PC, NAK                 ;SEND NAK
31
32                                          ;BUFFERS FULL
33
34 174024  132701  000001          B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
35 174030  001410                  BEQ     2$                              ;NO CK 1
36 174032  012702  010410          1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
37 174036  012703  010440                  MOV     #4384., R3              ;COM=SEL INBUF0
38 174042  012704  000001                  MOV     #BIT00, R4              ;WHICH CRC BIT
39 174046  004767  000026                  JSR     PC,     PACK            ;GO FIND PACKET
40 174052  132701  000002          2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
41 174056  001712                  BEQ     LIO                             ;NO LOOP BACK
42 174060  012702  010610                  MOV     #4488., R2              ;YES,COM=RDBUFADR
43 174064  012703  010640                  MOV     #4512., R3              ;COM=SEL INBUF1
44 174070  012704  000002                  MOV     #BIT01, R4              ;WHICH CRC BIT
45 174074  004767  000000                  JSR     PC,     PACK            ;GO FIND PACKET
```

```
                                                ;UNLOAD BUFFERS
                                                
174100  005010                  PACK:  CLR     (R0)                    ;CLEAR STATUS
174102  004767  000030                 JSR     PC,     EMBF            ;EMPTY BUFFER
174106  005710                        TST     (R0)                    ;GOOD UNLOAD
174110  100003                        BPL     4$                      ;YES
174112  004767  000450           3$:  JSR     PC,     NAK             ;NO NAK TO REPLACE
174116  000207                        RTS     PC                      ;RETURN
174120  004767  000226           4$:  JSR     PC,     LOAD            ;LOAD + CONTROL PARSE
174124  005710                        TST     (R0)                    ;CHECK RESPONSE
174126  100771                        BMI     3$                      ;BAD NAK
174130  004767  000424                 JSR     PC,     ACK             ;ACK IT AND CONTINUE
174134  000207                        RTS     PC
174136  012737  010600  172416 EMBF:  MOV     #4480., @#OPREG         ;RS (CRC)
174144  012737  002400  172416        MOV     #1280., @#OPREG         ;RS CMD
174152  013705  172416                MOV     @#IOBUF, R5             ;FETCH STATUS
174156  130405                        BITB    R4, R5                  ;CRC OK
174160  001002                        BNE     5$                      ;YES CONTINUE
174162  012710  177777                MOV     #-1,    (R0)            ;FLAG BAD CRC
174166  012737  160000  172410 5$:    MOV     #160000, @#BAR          ;ADDRESS TO WRITE
174174  010237  172416                MOV     R2,     @#OPREG         ;RDBUFADR
174200  012737  001400  172416        MOV     #768.,  @#OPREG         ;READ POINTER
174206  105737  172414                TSTB    @#CSR                   ;GOOD RD
174212  100375                        BPL     .-4                     ;NO LOOP UNTIL READY
174214  105037  172414                CLRB    @#CSR                   ;CLEAR DONE BIT
174220  013704  172416                MOV     @#IOBUF, R4             ;READ POINTER
174224  042704  177400                BIC     #177400, R4             ;CLEAR UNUSED BITS
174230  022704  000004                CMP     #4,     R4              ;CONTROL PACKET
174234  001405                        BEQ     6$                      ;YES UNLOAD
174236  022704  000204                CMP     #132.,  R4              ;MEMORY IMAGE UNLOAD
174242  001402                        BEQ     6$
174244  012710  177777                MOV     #-1,    (R0)            ;BAD UNLOAD
174250  005404                 6$:    NEG     R4                      ;2'S COMP
174252  010437  172412                MOV     R4,  @#WCR              ;BYTE COUNT
174256  010337  172416                MOV     R3,     @#OPREG         ;SEL BUFFER
174262  012737  001400  172416        MOV     #768.,  @#OPREG         ;FALSE RD
174270  105737  172414                TSTB    @#CSR                   ;GOOD RD
174274  100375                        BPL     .-4                     ;NO LOOP UNTIL READY
174276  105037  172414                CLRB    @#CSR
174302  012737  021000  172416        MOV     #8704., @#OPREG         ;FIRE DMA RD
174310  000240                        NOP                             ;DELAY
174312  105737  172414                TSTB    @#CSR                   ;GOOD DMA
174316  100402                        BMI     7$                      ;YES
174320  012710  177777                MOV     #-1,  (R0)              ;BAD DMA XFER
174324  105037  172414           7$:  CLRB    @#CSR                   ;CLEAR CSR
174330  012737  004400  172416        MOV     #2304., @#OPREG         ;RESET POINTER TO 255
174336  105737  172414                TSTB    @#CSR                   ;GOOD WD
174342  100375                        BPL     .-4                     ;NO RETRY
174344  105037  172414                CLRB    @#CSR                   ;CLEAR DONE BIT
174350  000207                 8$:    RTS     PC
                                       .ENABLE LSB
```

```
 1                                                  ;PACKET PARSE
 2
 3 174352  016702  000342          LOAD:  MOV     BUFFER, R2          ;BUFFER AREA
 4 174356  122712  000000                 CMPB    #0, (R2)            ;MEMORY OR CONTROL
 5 174362  001432                         BEQ     CMEQ                ;CONTROL PACKET
 6 174364  016703  000326                 MOV     PKCNT,  R3          ;PACKET COUNTER
 7 174370  022713  000144                 CMP     #100.,  (R3)        ;LIMIT
 8 174374  001003                         BNE     1$                  ;NO CONTINUE
 9 174376  005013                         CLR     (R3)                ;RESET POINTER
10 174400  005213                         INC     (R3)
11 174402  121312                 1$:     CMPB    (R3),   (R2)        ;NEW PACKET
12 174404  001401                         BEQ     2$
13 174406  000207                         RTS     PC                  ;RETURN
14 174410  122762  000007  000001  2$:     CMPB    #7,     1(R2)       ;TO BE PRINTED
15 174416  001444                         BEQ     10$
16 174420  062702  000002                 ADD     #2,     R2          ;OFFSET HEADER
17 174424  005213                         INC     (R3)                ;COUNT + 1
18 174426  017703  000262                 MOV     @MEMPT, R3
19 174432  012704  000100                 MOV     #64.,   R4          ;BYTE COUNT
20 174436  012223                 3$:     MOV     (R2)+, (R3)+        ;LOAD MEMORY
21 174440  077402                         SOB     R4,     3$          ;LOOP UNTIL ZERO
22 174442  010377  000246                 MOV     R3,     @MEMPT
23 174446  000207                         RTS     PC
24 174450  122762  000001  000001  CMEQ:  CMPB    #1,     1(R2)       ;IS IT A REPORT
25 174456  001004                         BNE     5$                  ;NO
26 174460  013703  174712                 MOV     @#STATUS, R3        ;LOCATION OF STATUS
27 174464  011310                         MOV     (R3),   (R0)        ;TRANSFER STATUS
28 174466  000207                         RTS     PC
29 174470  122762  000005  000001  5$:     CMPB    #5,     1(R2)       ;IS IT A START
30 174476  001005                         BNE     6$                  ;NO
31 174500  013700  000040                 MOV     @#40,   R0          ;FETCH START ADDRESS
32 174504  012706  001000                 MOV     #1000, SP           ;RESET USER STACK
33 174510  000110                         JMP     (R0)                ;LEAVE PROM TO RAM
34 174512  122762  000006  000001  6$:     CMPB    #6,     1(R2)       ;WAS IT A LOAD COMMAND
35 174520  001002                         BNE     7$                  ;NO
36 174522  000167  177136                 JMP     SRTLDR              ;YES GOTO LOAD START
37 174526  000207                 7$:     RTS     PC
38 174530  012704  000200         10$:    MOV     #128.,  R4          ;PRINT 200. BYTES
39 174534  062702  000002                 ADD     #2,     R2          ;OFFSET HEADER
40 174540  005213                         INC     (R3)                ;INC PCKET COUNTER
41 174542  105737  177564         11$:    TSTB    @#XCSR              ;PORT READY
42 174546  100375                         BPL     .-4                 ;NO WAIT UNTIL
43 174550  112237  177566                 MOVB    (R2)+,  @#XBUF      ;PRINT A BYTE
44 174554  077406                         SOB     R4,     11$         ;DO ALL 200
45 174556  000207                         RTS     PC
46
```

```
 1                                                  ;ACK/NAK ROUTINE
 2
 3 174560  012705  174670          ACK:    MOV     #ACKPK,   R5                ;ADDRESS OF ACK
 4 174564  000402                  8$:     BR      8$
 5 174566  012705  174700          NAK:    MOV     #NAKPK,   R5                ;ADDRESS OF NAK
 6 174572  012737  010540  172416 8$:     MOV     #4448.,  @#OPREG            ;SEL OUTBFØ
 7 174600  012504                          MOV     (R5)+,    R4                ;FETCH HEADER
 8 174602  067704  000110                  ADD     @PKCNT,   R4                ;ADD PACKET COUNT
 9 174606  010437  172416                  MOV     R4,      @#OPREG            ;WRITE IT
10 174612  105737  172414                  TSTB    @#CSR                       ;GOOD WD
11 174616  100375                          BPL     .-4                         ;NO LOOP
12 174620  012704  000003                  MOV     #3,       R4                ;BYTE COUNT
13 174624  012537  172416          9$:     MOV     (R5)+,  @#OPREG             ;WD
14 174630  105737  172414                  TSTB    @#CSR                       ;GOOD WD
15 174634  100375                          BPL     .-4                         ;NO LOOP
16 174636  077406                          SOB     R4,       9$                ;YES LOOP TIL Ø
17 174640  012737  010420  172416          MOV     #4368.,  @#OPREG            ;MODSTAT
18 174646  012737  004421  172416          MOV     #4421,   @#OPREG            ;SET BITS
19 174654  105737  172414                  TSTB    @#CSR                       ;GOOD WD
20 174660  100375                          BPL     .-4                         ;LOOP UNTIL
21 174662  105037  172414                  CLRB    @#CSR
22 174666  000207                          RTS     PC
23
24
25 174670  004400                  ACKPK:  .WORD   4400
26 174672  004402                          .WORD   4402
27 174674  004403                          .WORD   4403
28 174676  004405                          .WORD   4405
29 174700  004400                  NAKPK:  .WORD   4400
30 174702  004403                          .WORD   4403
31 174704  004403                          .WORD   4403
32 174706  004405                          .WORD   4405
33 174710  160400                  DATA:   .WORD   160400
34 174712  160402                  STATUS: .WORD   160402
35 174714  160404                  MEMPT:  .WORD   160404
36 174716  160406                  PKCNT:  .WORD   160406
37 174720  160000                  BUFFER: .WORD   160000
38 174722  004407                  NODE:   .WORD   4407
39
40         173000                          .END    START
```

```
PROM NODE 25   MACRO V03.01 27-MAR-79 18:56:53 PAGE 7-1
SYMBOL TABLE

ACK     174560          BIT07 = 000200          DATA    174710          MEMPT   174714          SRTLDR  173664
ACKPK   174670          BIT08 = 000400          EMBF    174136          NAK     174566          START   173000
BAR   = 172410          BIT14 = 040000          INBUF0  173276          NAKPK   174700          STAT    173564
BIT00 = 000001          BIT15 = 100000          INBUF1  173332          NODE    174722          STATUS  174712
BIT01 = 000002          BUFFER  174720          INIT    173070          OPREG = 172416          STRAM   173222
BIT02 = 000004          B01$    174024          INITLS  173014          OTBUF0  173366          WCR   = 172412
BIT03 = 000010          B2$     173750          IOBUF = 172416          OTBUF1  173402          XBUF  = 177566
BIT04 = 000020          B3$     173776          LIO     173704          PACK    174100          XCSR  = 177564
BIT05 = 000040          CMEQ    174450          LOAD    174352          PKCNT   174716          ZEROBP  173420
BIT06 = 000100          CSR   = 172414

. ABS.  174724      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM25=DK:PROM25
```

```
                                        .TITLE PROM NODE 26
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000

        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        177560                          SRCSR=  177560
        177562                          SRBUF=  177562
        177564                          SXCSR=  177564
        177566                          SXBUF=  177566
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1

173000  000005                  START:  RESET                   ;RESET BUS
173002  012700  000340                  MOV     #340,   R0      ;PRI=7
173006  106400                          MTPS    R0              ;CPU PRI
173010  012706  160500                  MOV     #160500, SP     ;RESET STACK

                                        ;INITIALIZE LINE SWITCHS

173014  012737  010420  172416  INITLS: MOV     #4368., @#OPREG ;MOD STAT
173022  012737  004414  172416          MOV     #4414,  @#OPREG ;SET SWITCHES
173030  105737  172414                  TSTB    @#CSR           ;GOOD WD
173034  100375                          BPL     .-4             ;LOOP UNTIL
173036  105037  172414                  CLRB    @#CSR
173042  012737  004540  172416          MOV     #4540,  @#OPREG ;RESET SWITCHS
173050  105737  172414                  TSTB    @#CSR           ;GOOD WD
173054  100375                          BPL     .-4             ;LOOP UNTIL
173056  105037  172414                  CLRB    @#CSR
173062  013703  175006                  MOV     @#STATUS, R3    ;LO CATION OF STATUS
173066  005005                          CLR     R5              ;CLEAR ERROR COUNT
```

```
 1                                                  ;INITIALIZE LIU
 2
 3 173070  005004                    INIT:   CLR     R4                      ;CLEAR LOOP COUNT
 4 173072  012737  010402  172416            MOV     #4354., @#OPREG         ;LDADR
 5 173100  012737  004400  172416            MOV     #2304., @#OPREG         ;ADDRESS=0
 6 173106  105737  172414                    TSTB    @#CSR                   ;GOOD WD
 7 173112  100375                            BPL     .-4                     ;NO RETRY
 8 173114  105037  172414                    CLRB    @#CSR
 9 173120  012737  010401  172416            MOV     #4353., @#OPREG         ;SEL ACRAM
10 173126  012700  000400                    MOV     #256., R0               ;COUNTER
11 173132  012737  004407  172416  1$:       MOV     #2311., @#OPREG         ;WRITE A NULL
12 173140  105737  172414                    TSTB    @#CSR                   ;GOOD WD
13 173144  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
14 173146  105037  172414                    CLRB    @#CSR
15 173152  077011                            SOB     R0,     1$              ;LOOP UNTIL ZERO
16 173154  012700  000400                    MOV     #256.,  R0              ;COUNTER
17 173160  012737  001400  172416  2$:       MOV     #768.,  @#OPREG         ;RD
18 173166  105737  172414                    TSTB    @#CSR                   ;GOOD RD
19 173172  100375                            BPL     .-4                     ;NO RETRY
20 173174  105037  172414                    CLRB    @#CSR
21 173200  013701  172416                    MOV     @#IOBUF, R1             ;FETCH DATA
22 173204  042701  177760                    BIC     #177760, R1             ;CLEAR BITS
23 173210  022701  000007                    CMP     #7,     R1              ;DATA=NULL
24 173214  001401                            BEQ     3$                      ;YES
25 173216  005204                            INC     R4                      ;NO REPORT IT
26 173220  077021                    3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO
27 173222  012737  010402  172416  STRAM:   MOV     #4354., @#OPREG         ;LDCTADR
28 173230  013737  175016  172416            MOV     @#NODE, @#OPREG         ;SEND ADDRESS
29 173236  105737  172414                    TSTB    @#CSR                   ;GOOD WRITE
30 173242  100375                            BPL     .-4                     ;LOOP UNTIL
31 173244  105037  172414                    CLRB    @#CSR
32 173250  012737  010401  172416            MOV     #4353., @#OPREG         ;SEL ACRAM
33 173256  012737  004404  172416            MOV     #4404,  @#OPREG         ;LOAD A 4
34 173264  105737  172414                    TSTB    @#CSR                   ;GOOD WRITE
35 173270  100375                            BPL     .-4                     ;NO LOOP UNTIL
36 173272  105037  172414                    CLRB    @#CSR
37 173276  012701  010410            INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
38 173302  012700  010440                    MOV     #10440, R0              ;SEL INBUF0
39 173306  004767  000106                    JSR     PC,     ZEROBP          ;SET POINTER=0
40 173312  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
41 173320  105737  172414                    TSTB    @#CSR                   ;GOOD READ
42 173324  100375                            BPL     .-4                     ;NO RETRY
43 173326  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
44 173332  012701  010610            INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
45 173336  012700  010640                    MOV     #10640, R0              ;SEL INBUF1 COMMAND
46 173342  004767  000052                    JSR     PC,     ZEROBP          ;POINTER=0
47 173346  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
48 173354  105737  172414                    TSTB    @#CSR                   ;GOOD RD
49 173360  100375                            BPL     .-4                     ;NO RETRY
50 173362  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
51 173366  012701  010510            OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
52 173372  012700  010540                    MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
53 173376  004767  000016                    JSR     PC,     ZEROBP
54 173402  012701  010710            OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
55 173406  012700  010740                    MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
56 173412  004767  000002                    JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
57 173416  000462                            BR      STAT                    ;GO CLEAR STATUS
```

```
                                        ;INITIALIZE BUFFERS & STATUS
                                        
173420  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
173424  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173432  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173436  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173440  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173444  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
173450  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173454  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
173460  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
173464  001412                          BEQ     5$
173466  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
173474  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173500  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173502  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173506  005302                          DEC     R2                      ;POINTER-1
173510  000763                          BR      4$
173512  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
173516  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173524  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173530  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173532  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173536  013702  172416                  MOV     @#IOBUF, R2             ;FETCH NEW POINTER
173542  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173546  022702  000000                  CMP     #0,     R2              ;POINTER=0
173552  001401                          BEQ     6$
173554  005204                          INC     R4                      ;REPORT ERROR
173556  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
173562  000207                          RTS     PC                      ;RETURN
173564  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR:RS
173572  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
173600  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
173606  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
173612  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
173616  022700  000000                  CMP     #0,     R0              ;STATUS=0 ?
173622  001401                          BEQ     7$
173624  005204                          INC     R4                      ;STATUS BAD REPORT
173626  022704  000000          7$:     CMP     #0,     R4              ;WERE THERE ERRORS
173632  001406                          BEQ     8$                      ;NO GOOD INIT
173634  005205                          INC     R5                      ;ERROR +1
173636  022705  000005                  CMP     #5,     R5              ;ERROR LIMIT YET
173642  001406                          BEQ     9$                      ;YES RETURN A 1(ERROR)
173644  000167  177220                  JMP     INIT                    ;NO RESTART VINIT
173650  012713  000001          8$:     MOV     #1,     (R3)            ;GOOD INIT.
173654  000167  000004                  JMP     SRTLDR
173660  012713  177777          9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
                                        .ENABLE LSB
```

```
 1                                          ;LOADER START
 2
 3 173664  005077  001120          SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
 4 173670  005077  001116                  CLR     @PKCNT                  ;PACKET COUNTER
 5 173674  005277  001112                  INC     @PKCNT                  ;START'S AT 1
 6 173700  105037  172414                  CLRB    @#CSR                   ;CLEAR CSR
 7 173704  005737  172414          LIO:    TST     @#CSR                   ;REQUEST YET
 8 173710  100375                          BPL     .-4                     ;NO LOOP ON LIO
 9 173712  013700  175004                  MOV     @#DATA,  R0             ;ADDRESS OF DATA
10 173716  005010                          CLR     (R0)                    ;CLEAR DATA
11 173720  012706  160500                  MOV     #160500,  SP            ;RESET SP
12 173724  012737  010400  172416          MOV     #4352., @#OPREG         ;WCR : RS(0)
13 173732  012737  002400  172416          MOV     #1280., @#OPREG         ;RD
14 173740  013701  172416                  MOV     @#IGBUF, R1
15 173744  042701  177400                  BIC     #177400, R1             ;CLEAR UNUSED BITS
16
17                                          ;BUFFER OVERFLOW
18
19 173750  132701  000004          B2$:    BITB    #BIT02, R1              ;OV-FL
20 173754  001410                          BEQ     B3$
21 173756  012702  010410                  MOV     #4360., R2              ;COM=SEL INBUF 0
22 173762  012703  010440                  MOV     #4384., R3              ;COM=RDBUFPNT
23 173766  004767  000144                  JSR     PC, EMBF                ;EMPTY BUFFER
24 173772  004767  000664                  JSR     PC, NAK                 ;SEND NAK
25 173776  132701  000010          B3$:    BITB    #BIT03, R1              ;OV-FL
26 174002  001410                          BEQ     B01$
27 174004  012702  010610                  MOV     #4488., R2              ;COM=SEL INBUF 1,
28 174010  012703  010640                  MOV     #4512., R3              ;COM=RDBUFPNT
29 174014  004767  000116                  JSR     PC, EMBF
30 174020  004767  000636                  JSR     PC, NAK                 ;SEND NAK
31
32                                          ;BUFFERS FULL
33
34 174024  132701  000001          B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
35 174030  001410                          BEQ     2$                      ;NO CK 1
36 174032  012702  010410          1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
37 174036  012703  010440                  MOV     #4384., R3              ;COM=SEL INBUF0
38 174042  012704  000001                  MOV     #BIT00, R4              ;WHICH CRC BIT
39 174046  004767  000026                  JSR     PC,     PACK            ;GO FIND PACKET
40 174052  132701  000002          2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
41 174056  001712                          BEQ     LIO                     ;NO LOOP BACK
42 174060  012702  010610                  MOV     #4488., R2              ;YES,COM=RDBUFADR
43 174064  012703  010640                  MOV     #4512., R3              ;COM=SEL INBUF1
44 174070  012704  000002                  MOV     #BIT01, R4              ;WHICH CRC BIT
45 174074  004767  000000                  JSR     PC,     PACK            ;GO FIND PACKET
```

```
                                                ;UNLOAD BUFFERS
                                                
174100  005010                  PACK:   CLR     (R0)            ;CLEAR STATUS
174102  004767  000030                  JSR     PC,     EMBF    ;EMPTY BUFFER
174106  005710                          TST     (R0)            ;GOOD UNLOAD
174110  100003                          BPL     4$              ;YES
174112  004767  000544          3$:     JSR     PC,     NAK     ;NO NAK TO REPLACE
174116  000207                          RTS     PC              ;RETURN
174120  004767  000226          4$:     JSR     PC,     LOAD    ;LOAD + CONTROL PARSE
174124  005710                          TST     (R0)            ;CHECK RESPONSE
174126  100771                          BMI     3$              ;BAD NAK
174130  004767  000520                  JSR     PC,     ACK     ;ACK IT AND CONTINUE
174134  000207                          RTS     PC
174136  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG ;RS (CRC)
174144  012737  002400  172416          MOV     #1280., @#OPREG ;RS CMD
174152  013705  172416                  MOV     @#IOBUF, R5     ;FETCH STATUS
174156  130405                          BITB    R4, R5          ;CRC OK
174160  001002                          BNE     5$              ;YES CONTINUE
174162  012710  177777                  MOV     #-1,    (R0)    ;FLAG BAD CRC
174166  012737  160000  172410  5$:     MOV     #160000, @#BAR  ;ADDRESS TO WRITE
174174  010237  172416                  MOV     R2,     @#OPREG ;RDBUFADR
174200  012737  001400  172416          MOV     #768.,  @#OPREG ;READ POINTER
174206  105737  172414                  TSTB    @#CSR           ;GOOD RD
174212  100375                          BPL     .-4             ;NO LOOP UNTIL READY
174214  105037  172414                  CLRB    @#CSR           ;CLEAR DONE BIT
174220  013704  172416                  MOV     @#IOBUF, R4     ;READ POINTER
174224  042704  177400                  BIC     #177400, R4     ;CLEAR UNUSED BITS
174230  022704  000004                  CMP     #4,     R4      ;CONTROL PACKET
174234  001405                          BEQ     6$              ;YES UNLOAD
174236  022704  000204                  CMP     #132.,  R4      ;MEMORY IMAGE UNLOAD
174242  001402                          BEQ     6$
174244  012710  177777                  MOV     #-1,    (R0)    ;BAD UNLOAD
174250  005404                  6$:     NEG     R4              ;2'S COMP
174252  010437  172412                  MOV     R4,  @#WCR      ;BYTE COUNT
174256  010337  172416                  MOV     R3,     @#OPREG ;SEL BUFFER
174262  012737  001400  172416          MOV     #768.,  @#OPREG ;FALSE RD
174270  105737  172414                  TSTB    @#CSR           ;GOOD RD
174274  100375                          BPL     .-4             ;NO LOOP UNTIL READY
174276  105037  172414                  CLRB    @#CSR
174302  012737  021000  172416          MOV     #8704., @#OPREG ;FIRE DMA RD
174310  000240                          NOP                     ;DELAY
174312  105737  172414                  TSTB    @#CSR           ;GOOD DMA
174316  100402                          BMI     7$              ;YES
174320  012710  177777                  MOV     #-1,  (R0)      ;BAD DMA XFER
174324  105037  172414          7$:     CLRB    @#CSR           ;CLEAR CSR
174330  012737  004400  172416          MOV     #2304., @#OPREG ;RESET POINTER TO 255
174336  105737  172414                  TSTB    @#CSR           ;GOOD WD
174342  100375                          BPL     .-4             ;NO RETRY
174344  105037  172414                  CLRB    @#CSR           ;CLEAR DONE BIT
174350  000207                  8$:     RTS     PC
                                        .ENABLE LSB
```

```
 1                                                  ;PACKET PARSE
 2
 3 174352  016702  000436          LOAD:   MOV     BUFFER, R2              ;BUFFER AREA
 4 174356  122712  000000                  CMPB    #0, (R2)                ;MEMORY OR CONTROL
 5 174362  001432                          BEQ     CMEQ                    ;CONTROL PACKET
 6 174364  016703  000422                  MOV     PKCNT,  R3              ;PACKET COUNTER
 7 174370  022713  000144                  CMP     #100.,  (R3)            ;LIMIT
 8 174374  001002                          BNE     1$                      ;NO CONTINUE
 9 174376  005013                          CLR     (R3)                    ;RESET POINTER
10 174400  005213                          INC     (R3)
11 174402  121312                  1$:     CMPB    (R3),   (R2)            ;NEW PACKET
12 174404  001401                          BEQ     2$
13 174406  000207                          RTS     PC                      ;RETURN
14 174410  122762  000007  000001  2$:     CMPB    #7,     1(R2)           ;IS IT FOR SIG
15 174416  001501                          BEQ     10$                     ;YES BR
16 174420  062702  000002                  ADD     #2,     R2              ;OFFSET HEADER
17 174424  005213                          INC     (R3)                    ;COUNT + 1
18 174426  017703  000356                  MOV     @MEMPT, R3
19 174432  012704  000100                  MOV     #64.,   R4              ;BYTE COUNT
20 174436  012223                  3$:     MOV     (R2)+, (R3)+            ;LOAD MEMORY
21 174440  077402                          SOB     R4,     3$              ;LOOP UNTIL ZERO
22 174442  010377  000342                  MOV     R3,     @MEMPT
23 174446  000207                          RTS     PC
24 174450  122762  000001  000001  CMEQ:   CMPB    #1,     1(R2)           ;IS IT A REPORT
25 174456  001004                          BNE     5$                      ;NO
26 174460  013703  175006                  MOV     @#STATUS, R3            ;LOCATION OF STATUS
27 174464  011310                          MOV     (R3),   (R0)            ;TRANSFER STATUS
28 174466  000207                          RTS     PC
29 174470  122762  000005  000001  5$:     CMPB    #5,     1(R2)           ;IS IT A START
30 174476  001005                          BNE     6$                      ;NO
31 174500  013700  000040                  MOV     @#40,   R0              ;FETCH START ADDRESS
32 174504  012706  001000                  MOV     #1000, SP               ;RESET USER STACK
33 174510  000110                          JMP     (R0)                    ;LEAVE PROM TO RAM
34 174512  122762  000006  000001  6$:     CMPB    #6,     1(R2)           ;WAS IT A LOAD COMMAND
35 174520  001003                          BNE     8$                      ;NO
36 174522  000167  177136                  JMP     SRTIDR                  ;YES GOTO LOAD START
37 174526  000207                  7$:     RTS     PC
38
39
```

```
                                        ;SIG LOADER
                                        
174530  122762  000010  000001  8$:   CMPB    #8.,    1(R2)           ;INTERFACE STATUS
174536  001016                        BNE     9$
174540  105737  177560                TSTB    @#SRCSR                 ;BYTE THERE
174544  100402                        BMI     12$
174546  000167  177112                JMP     SRTLDR                  ;SIMULATES NO RESPONSE
174552  113702  177562        12$:    MOVB    @#SRBUF, R2             ;FETCH STATUS
174556  042702  177400                BIC     #177400, R2             ;CLEAR MST
174562  122702  000122                CMPB    #122,    R2             ;IS IT READY
174566  001757                        BEQ     7$                      ;YES IT IS
174570  000167  177070                JMP     SRTLDR                  ;SIMULATE A NO RESPONSE
174574  122762  000011  000001  9$:   CMPB    #9.,    1(R2)           ;LOAD MODE INTERFACE
174602  001351                        BNE     7$                      ;NO DISREGUARD
174604  105737  177564                TSTB    @#SICSR                 ;INTERFACE TO SEND
174610  100375                        BPL     .-4
174612  112737  000114  177566        MOVB    #114,   @#SIBUF         ;SEND LOAD COMMAND
174620  000742                        BR      7$                      ;RETURN
174622  062702  000002        10$:    ADD     #2,     R2              ;OFFSET HEADER
174626  005213                        INC     (R3)                    ;PACKET COUNT+1
174630  012704  000200                MOV     #128.,  R4              ;BYTE COUNT
174634  105737  177564        11$:    TSTB    @#SICSR                 ;INTERFACE READY
174640  100375                        BPL     .-4                     ;NO LOOP UNTIL
174642  112237  177566                MOVB    (R2)+,  @#SIBUF         ;SEND BYTE
174646  000240                        NOP                             ;DELAY
174650  077407                        SOB     R4,     11$             ;LOOP UNTIL ALL SENT
174652  000207                        RTS     PC
```

```
 1                                          .ENABLE LSB
 2                                          ;ACK/NAK ROUTINE
 3
 4 174654  012705  174764          ACK:    MOV     #ACKPK,   R5          ;ADDRESS OF ACK
 5 174660  000402                  1$      BR      1$
 6 174662  012705  174774          NAK:    MOV     #NAKPK,   R5          ;ADDRESS OF NAK
 7 174666  012737  010540  172416  1$:     MOV     #4448., @#OPREG       ;SEL OUTBF0
 8 174674  012504                          MOV     (R5)+,  R4            ;FETCH HEADER
 9 174676  067704  000110                  ADD     @PKCNT, R4            ;ADD PACKET COUNT
10 174702  010437  172416                  MOV     R4,     @#OPREG       ;WRITE IT
11 174706  105737  172414                  TSTB    @#CSR                 ;GOOD WD
12 174712  100375                          BPL     .-4                   ;NO LOOP
13 174714  012704  000003                  MOV     #3,     R4            ;BYTE COUNT
14 174720  012537  172416          2$:     MOV     (R5)+, @#OPREG        ;WD
15 174724  105737  172414                  TSTB    @#CSR                 ;GOOD WD
16 174730  100375                          BPL     .-4                   ;NO LOOP
17 174732  077406                          SOB     R4,     2$            ;YES LOOP TIL 0
18 174734  012737  010420  172416          MOV     #4368., @#OPREG       ;MODSTAT
19 174742  012737  004421  172416          MOV     #4421,  @#OPREG       ;SET BITS
20 174750  105737  172414                  TSTB    @#CSR                 ;GOOD WD
21 174754  100375                          BPL     .-4                   ;LOOP UNTIL
22 174756  105037  172414                  CLRB    @#CSR
23 174762  000207                          RTS     PC
24
25
26 174764  004400          ACKPK:  .WORD   4400
27 174766  004402                  .WORD   4402
28 174770  004403                  .WORD   4403
29 174772  004405                  .WORD   4405
30 174774  004400          NAKPK:  .WORD   4400
31 174776  004403                  .WORD   4403
32 175000  004403                  .WORD   4403
33 175002  004405                  .WORD   4405
34 175004  160400          DATA:   .WORD   160400
35 175006  160402          STATUS: .WORD   160402
36 175010  160404          MEMPT:  .WORD   160404
37 175012  160406          PKCNT:  .WORD   160406
38 175014  160000          BUFFER: .WORD   160000
39 175016  004410          NODE:   .WORD   4410
40
41         173000                       .END    START
```

```
FROM NODE 26   MACRO V03.01 27-MAR-79 18:57:45 PAGE 8-1
SYMBOL TABLE

ACK     174654          BIT07 = 000200          DATA    175004          NAK     174662          SRTLDR  173664
ACKPK   174764          BIT08 = 000400          EMBF    174136          NAKPK   174774          START   173000
BAR   = 172410          BIT14 = 040000          INBUF0  173276          NODE    175016          STAT    173564
BIT00 = 000001          BIT15 = 100000          INBUF1  173332          OPREG = 172416          STATUS  175006
BIT01 = 000002          BUFFER  175014          INIT    173070          OTBUF0  173366          STRAM   173222
BIT02 = 000004          B01$    174024          INITLS  173014          OTBUF1  173402          SXBUF = 177566
BIT03 = 000010          B2$     173750          IOBUF = 172416          PACK    174100          SXCSR = 177564
BIT04 = 000020          B3$     173776          LIO     173704          PKCNT   175012          WCR   = 172412
BIT05 = 000040          CMEQ    174450          LOAD    174352          SRBUF = 177562          ZEROBP  173420
BIT06 = 000100          CSR   = 172414          MEMPT   175010          SRCSR = 177560

. ABS.  175020     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM26=DK:PROM26
```

```
                                        .TITLE PROM NODE 27
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000

        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1

173000  000005                  START:  RESET                   ;RESET BUS
173002  012700  000340                  MOV     #340,   R0      ;PRI=7
173006  106400                          MTPS    R0              ;CPU PRI
173010  012706  160500                  MOV     #160500, SP     ;RESET STACK

                                        ;INITIALIZE LINE SWITCHS

173014  012737  010420  172416  INITLS: MOV     #4368., @#OPREG ;MOD STAT
173022  012737  004414  172416          MOV     #4414,  @#OPREG ;SET SWITCHES
173030  105737  172414                  TSTB    @#CSR           ;GOOD WD
173034  100375                          BPL     .-4             ;LOOP UNTIL
173036  105037  172414                  CLRB    @#CSR
173042  012737  004540  172416          MOV     #4540,  @#OPREG ;RESET SWITCHS
173050  105737  172414                  TSTB    @#CSR           ;GOOD WD
173054  100375                          BPL     .-4             ;LOOP UNTIL
173056  105037  172414                  CLRB    @#CSR
173062  013703  174652                  MOV     @#STATUS, R3    ;LC ATION OF STATUS
173066  005005                          CLR     R5              ;CLEAR ERROR COUNT
```

```
                                        ;INITIALIZE LIU

173070  005004                  INIT:   CLR     R4                      ;CLEAR LOOP COUNT
173072  012737  010402  172416          MOV     #4354., @#OPREG         ;LDADR
173100  012737  004400  172416          MOV     #2304., @#OPREG         ;ADDRESS=0
173106  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173112  100375                          BPL     .-4                     ;NO RETRY
173114  105037  172414                  CLRB    @#CSR
173120  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
173126  012700  000400                  MOV     #256., R0               ;COUNTER
173132  012737  004407  172416  1$:     MOV     #2311., @#OPREG         ;WRITE A NULL
173140  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173144  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173146  105037  172414                  CLRB    @#CSR
173152  077011                          SOB     R0,     1$              ;LOOP UNTIL ZERO
173154  012700  000400                  MOV     #256.,  R0              ;COUNTER
173160  012737  001400  172416  2$:     MOV     #768.,  @#OPREG         ;RD
173166  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173172  100375                          BPL     .-4                     ;NO RETRY
173174  105037  172414                  CLRB    @#CSR
173200  013701  172416                  MOV     @#IOBUF, R1             ;FETCH DATA
173204  042701  177760                  BIC     #177760, R1             ;CLEAR BITS
173210  022701  000007                  CMP     #7,      R1             ;DATA=NULL
173214  001401                          BEQ     3$                      ;YES
173216  005204                          INC     R4                      ;NO REPORT IT
173220  077021                  3$:     SOB     R0,     2$              ;LOOP UNTIL ZERO
173222  012737  010402  172416  STRAM:  MOV     #4354., @#OPREG         ;LDCTADR
173230  013737  174662  172416          MOV     @#NODE, @#OPREG         ;SEND ADDRESS
173236  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
173242  100375                          BPL     .-4                     ;LOOP UNTIL
173244  105037  172414                  CLRB    @#CSR
173250  012737  010401  172416          MOV     #4353., @#OPREG         ;SEL ACRAM
173256  012737  004404  172416          MOV     #4404,  @#OPREG         ;LOAD A 4
173264  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
173270  100375                          BPL     .-4                     ;NO LOOP UNTIL
173272  105037  172414                  CLRB    @#CSR
173276  012701  010410          INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
173302  012700  010440                  MOV     #10440, R0              ;SEL INBUFO
173306  004767  000106                  JSR     PC,     ZEROBP          ;SET POINTER=0
173312  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
173320  105737  172414                  TSTB    @#CSR                   ;GOOD READ
173324  100375                          BPL     .-4                     ;NO RETRY
173326  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173332  012701  010610          INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
173336  012700  010640                  MOV     #10640, R0              ;SEL INBUF1 COMMAND
173342  004767  000052                  JSR     PC,     ZEROBP          ;POINTER=0
173346  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
173354  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173360  100375                          BPL     .-4                     ;NO RETRY
173362  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173366  012701  010510          OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
173372  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
173376  004767  000016                  JSR     PC,     ZEROBP
173402  012701  010710          OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
173406  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
173412  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
173416  000462                          BR      STAT                    ;GO CLEAR STATUS
```

```
                                        ;INITIALIZE BUFFERS & STATUS
                                        
173420  010137  172416          ZEROBF: MOV     R1,     @#OPREG         ;RDBUFADR
173424  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173432  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173436  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173440  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173444  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
173450  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173454  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
173460  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
173464  001412                          BEQ     5$
173466  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
173474  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173500  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173502  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173506  005302                          DEC     R2                      ;POINTER-1
173510  000763                          BR      4$
173512  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
173516  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
173524  105737  172414                  TSTB    @#CSR                   ;GOOD RD
173530  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
173532  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
173536  013702  172416                  MOV     @#IOBUF, R2             ;FETCH NEW POINTER
173542  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
173546  022702  000000                  CMP     #0,     R2              ;POINTER=0
173552  001401                          BEQ     6$
173554  005204                          INC     R4                      ;REPORT ERROR
173556  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
173562  000207                          RTS     PC                      ;RETURN
173564  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR:RS
173572  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
173600  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
173606  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
173612  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
173616  022700  000000                  CMP     #0,     R0              ;STATUS=0 ?
173622  001401                          BEQ     7$
173624  005204                          INC     R4                      ;STATUS BAD REPORT
173626  022704  000000          7$:     CMP     #0,     R4              ;WERE THERE ERRORS
173632  001406                          BEQ     8$                      ;NO GOOD INIT
173634  005205                          INC     R5                      ;ERROR +1
173636  022705  000005                  CMP     #5,     R5              ;ERROR LIMIT YET
173642  001406                          BEQ     9$                      ;YES RETURN A 1(ERROR)
173644  000167  177220                  JMP     INIT                    ;NO RESTART YINIT
173650  012713  000001          8$:     MOV     #1,     (R3)            ;GOOD INIT.
173654  000167  000004                  JMP     SRTLDR
173660  012713  177777          9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
                                        .ENABLE LSB
```

```
                                        ;LOADER START
                                        
173664  005077  000764        SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
173670  005077  000762                CLR     @PKCNT                  ;PACKET COUNTER
173674  005277  000756                INC     @PKCNT                  ;START'S AT 1
173700  105037  172414                CLRB    @#CSR                   ;CLEAR CSR
173704  005737  172414        LIO:    TST     @#CSR                   ;REQUEST YET
173710  100375                      BPL     .-4                     ;NO LOOP ON LIO
173712  013700  174650                MOV     @#DATA,  R0             ;ADDRESS OF DATA
173716  005010                      CLR     (R0)                    ;CLEAR DATA
173720  012706  160500                MOV     #160500,  SP            ;RESET SP
173724  012737  010400  172416        MOV     #4352., @#OPREG         ;WCR : RS(0)
173732  012737  002400  172416        MOV     #1280., @#OPREG         ;RD
173740  013701  172416                MOV     @#IOBUF, R1
173744  042701  177400                BIC     #177400, R1             ;CLEAR UNUSED BITS
                                        
                                        ;BUFFER OVERFLOW
                                        
173750  132701  000004        B2$:    BITB    #BIT02, R1              ;OV-FL
173754  001410                      BEQ     B3$
173756  012702  010410                MOV     #4360., R2              ;COM=SEL INBUF 0
173762  012703  010440                MOV     #4384., R3              ;COM=RDBUFPNT
173766  004767  000144                JSR     PC, EMBF                ;EMPTY BUFFER
173772  004767  000530                JSR     PC, NAK                 ;SEND NAK
173776  132701  000010        B3$:    BITB    #BIT03, R1              ;OV-FL
174002  001410                      BEQ     B01$
174004  012702  010610                MOV     #4488., R2              ;COM=SEL INBUF 1
174010  012703  010640                MOV     #4512., R3              ;COM=RDBUFPNT
174014  004767  000116                JSR     PC, EMBF
174020  004767  000502                JSR     PC, NAK                 ;SEND NAK
                                        
                                        ;BUFFERS FULL
                                        
174024  132701  000001        B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
174030  001410                      BEQ     2$                      ;NO CK 1
174032  012702  010410        1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
174036  012703  010440                MOV     #4384., R3              ;COM=SEL INBUF0
174042  012704  000001                MOV     #BIT00, R4              ;WHICH CRC BIT
174046  004767  000026                JSR     PC,     PACK            ;GO FIND PACKET
174052  132701  000002        2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
174056  001712                      BEQ     LIO                     ;NO LOOP BACK
174060  012702  010610                MOV     #4488., R2              ;YES,COM=RDBUFADR
174064  012703  010640                MOV     #4512., R3              ;COM=SEL INBUF1
174070  012704  000002                MOV     #BIT01, R4              ;WHICH CRC BIT
174074  004767  000000                JSR     PC,     PACK            ;GO FIND PACKET
```

```
 1                                          ;UNLOAD BUFFERS
 2
 3 174100  005010                  PACK:    CLR     (R0)                ;CLEAR STATUS
 4 174102  004767  000030                   JSR     PC,     EMBF        ;EMPTY BUFFER
 5 174106  005710                           TST     (R0)                ;GOOD UNLOAD
 6 174110  100003                           BPL     4$                  ;YES
 7 174112  004767  000410          3$:      JSR     PC,     NAK         ;NO NAK TO REPLACE
 8 174116  000207                           RTS     PC                  ;RETURN
 9 174120  004767  000226          4$:      JSR     PC,     LOAD        ;LOAD + CONTROL PARSE
10 174124  005710                           TST     (R0)                ;CHECK RESPONSE
11 174126  100771                           BMI     3$                  ;BAD NAK
12 174130  004767  000364                   JSR     PC,     ACK         ;ACK IT AND CONTINUE
13 174134  000207                           RTS     PC
14 174136  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG     ;RS (CRC)
15 174144  012737  002400  172416           MOV     #1280., @#OPREG     ;RS CMD
16 174152  013705  172416                   MOV     @#IOBUF, R5         ;FETCH STATUS
17 174156  130405                           BITB    R4, R5              ;CRC OK
18 174160  001002                           BNE     5$                  ;YES CONTINUE
19 174162  012710  177777                   MOV     #-1,    (R0)        ;FLAG BAD CRC
20 174166  012737  160000  172410  5$:      MOV     #160000, @#BAR      ;ADDRESS TO WRITE
21 174174  010237  172416                   MOV     R2,     @#OPREG     ;RDBUFADR
22 174200  012737  001400  172416           MOV     #768.,  @#OPREG     ;READ POINTER
23 174206  105737  172414                   TSTB    @#CSR               ;GOOD RD
24 174212  100375                           BPL     .-4                 ;NO LOOP UNTIL READY
25 174214  105037  172414                   CLRB    @#CSR               ;CLEAR DONE BIT
26 174220  013704  172416                   MOV     @#IOBUF, R4         ;READ POINTER
27 174224  042704  177400                   BIC     #177400, R4         ;CLEAR UNUSED BITS
28 174230  022704  000004                   CMP     #4,     R4          ;CONTROL PACKET
29 174234  001405                           BEQ     6$                  ;YES UNLOAD
30 174236  022704  000204                   CMP     #132.,  R4          ;MEMORY IMAGE UNLOAD
31 174242  001402                           BEQ     6$
32 174244  012710  177777                   MOV     #-1,    (R0)        ;BAD UNLOAD
33 174250  005404                  6$:      NEG     R4                  ;2'S COMP
34 174252  010437  172412                   MOV     R4,  @#WCR          ;BYTE COUNT
35 174256  010337  172416                   MOV     R3,     @#OPREG     ;SEL BUFFER
36 174262  012737  001400  172416           MOV     #768.,  @#OPREG     ;FALSE RD
37 174270  105737  172414                   TSTB    @#CSR               ;GOOD RD
38 174274  100375                           BPL     .-4                 ;NO LOOP UNTIL READY
39 174276  105037  172414                   CLRB    @#CSR
40 174302  012737  021000  172416           MOV     #8704., @#OPREG     ;FIRE DMA RD
41 174310  000240                           NOP                         ;DELAY
42 174312  105737  172414                   TSTB    @#CSR               ;GOOD DMA
43 174316  100402                           BMI     7$                  ;YES
44 174320  012710  177777                   MOV     #-1,  (R0)          ;BAD DMA XFER
45 174324  105037  172414          7$:      CLRB    @#CSR               ;CLEAR CSR
46 174330  012737  004400  172416           MOV     #2304., @#OPREG     ;RESET POINTER TO 255
47 174336  105737  172414                   TSTB    @#CSR               ;GOOD WD
48 174342  100375                           BPL     .-4                 ;NO RETRY
49 174344  105037  172414                   CLRB    @#CSR               ;CLEAR DONE BIT
50 174350  000207                  8$:      RTS     PC
51                                          .ENABLE LSB
```

```
 1                                          ;PACKET PARSE
 2
 3 174352  016702  000302          LOAD:   MOV     BUFFER, R2          ;BUFFER AREA
 4 174356  122712  000000                  CMPB    #0, (R2)            ;MEMORY OR CONTROL
 5 174362  001426                          BEQ     CMEQ                ;CONTROL PACKET
 6 174364  016703  000266                  MOV     PKCNT,  R3          ;PACKET COUNTER
 7 174370  022713  000144                  CMP     #100.,  (R3)        ;LIMIT
 8 174374  001002                          BNE     1$                  ;NO CONTINUE
 9 174376  005013                          CLR     (R3)                ;RESET POINTER
10 174400  005213                          INC     (R3)
11 174402  121312                  1$:     CMPB    (R3),   (R2)        ;NEW PACKET
12 174404  001401                          BEQ     2$
13 174406  000207                          RTS     PC                  ;RETURN
14 174410  062702  000002          2$:     ADD     #2,     R2          ;OFFSET HEADER
15 174414  005213                          INC     (R3)                ;COUNT + 1
16 174416  017703  000232                  MOV     @MEMPT, R3
17 174422  012704  000100                  MOV     #64.,   R4          ;BYTE COUNT
18 174426  012223                  3$:     MOV     (R2)+, (R3)+        ;LOAD MEMORY
19 174430  077402                          SOB     R4,     3$          ;LOOP UNTIL ZERO
20 174432  010377  000216                  MOV     R3,     @MEMPT
21 174436  000207                          RTS     PC
22 174440  122762  000001  000001  CMEQ:   CMPB    #1,     1(R2)       ;IS IT A REPORT
23 174446  001004                          BNE     5$                  ;NO
24 174450  013703  174652                  MOV     @#STATUS, R3        ;LOCATION OF STATUS
25 174454  011310                          MOV     (R3),   (R0)        ;TRANSFER STATUS
26 174456  000207                          RTS     PC
27 174460  122762  000005  000001  5$:     CMPB    #5,     1(R2)       ;IS IT A START
28 174466  001005                          BNE     6$                  ;NO
29 174470  013700  000040                  MOV     @#40,   R0          ;FETCH START ADDRESS
30 174474  012706  001000                  MOV     #1000, SP           ;RESET USER STACK
31 174500  000110                          JMP     (R0)                ;LEAVE PROM TO RAM
32 174502  122762  000006  000001  6$:     CMPB    #6,     1(R2)       ;WAS IT A LOAD COMMAND
33 174510  001002                          BNE     7$                  ;NO
34 174512  000167  177146                  JMP     SRTLDR              ;YES GOTO LOAD START
35 174516  000207                  7$:     RTS     PC
36
```

```
 1                                              ;ACK/NAK ROUTINE
 2
 3 174520  012705  174630          ACK:  MOV    #ACKPK,   R5          ;ADDRESS OF ACK
 4 174524  000402                        BR     8$
 5 174526  012705  174640          NAK:  MOV    #NAKPK,   R5          ;ADDRESS OF NAK
 6 174532  012737  010540  172416  8$:   MOV    #4448., @#OPREG       ;SEL OUTBF0
 7 174540  012504                        MOV    (R5)+,  R4            ;FETCH HEADER
 8 174542  067704  000110                ADD    @PKCNT, R4            ;ADD PACKET COUNT
 9 174546  010437  172416                MOV    R4,     @#OPREG       ;WRITE IT
10 174552  105737  172414                TSTB   @#CSR                 ;GOOD WD
11 174556  100375                        BPL    .-4                   ;NO LOOP
12 174560  012704  000003                MOV    #3,     R4            ;BYTE COUNT
13 174564  012537  172416          9$:   MOV    (R5)+, @#OPREG        ;WD
14 174570  105737  172414                TSTB   @#CSR                 ;GOOD WD
15 174574  100375                        BPL    .-4                   ;NO LOOP
16 174576  077406                        SOB    R4,     9$            ;YES LOOP TIL 0
17 174600  012737  010420  172416        MOV    #4368., @#OPREG       ;MODSTAT
18 174606  012737  004421  172416        MOV    #4421,  @#OPREG       ;SET BITS
19 174614  105737  172414                TSTB   @#CSR                 ;GOOD WD
20 174620  100375                        BPL    .-4                   ;LOOP UNTIL
21 174622  105037  172414                CLRB   @#CSR
22 174626  000207                        RTS    PC
23
24
25 174630  004400                ACKPK:  .WORD  4400
26 174632  004402                        .WORD  4402
27 174634  004403                        .WORD  4403
28 174636  004405                        .WORD  4405
29 174640  004400                NAKPK:  .WORD  4400
30 174642  004403                        .WORD  4403
31 174644  004403                        .WORD  4403
32 174646  004405                        .WORD  4405
33 174650  160400                DATA:   .WORD  160400
34 174652  160402                STATUS: .WORD  160402
35 174654  160404                MEMPT:  .WORD  160404
36 174656  160406                PKCNT:  .WORD  160406
37 174660  160000                BUFFER: .WORD  160000
38 174662  004411                NODE:   .WORD  4411
39
40         173000                        .END   START
```

```
PROM NODE 27   MACRO V03.01 27-MAR-79 18:58:44 PAGE 7-1
SYMBOL TABLE

ACK     174520          BIT06 = 000100          CMEQ    174440          LIO     173704          PACK    174100
ACKPK   174630          BIT07 = 000200          CSR   = 172414          LOAD    174352          PKCNT   174656
BAR   = 172410          BIT08 = 000400          DATA    174650          MEMPT   174654          SRTLDR  173664
BIT00 = 000001          BIT14 = 040000          EMBF    174136          NAK     174526          START   173000
BIT01 = 000002          BIT15 = 100000          INBUF0  173276          NAKPK   174640          STAT    173564
BIT02 = 000004          BUFFER  174660          INBUF1  173332          NODE    174662          STATUS  174652
BIT03 = 000010          B01$    174024          INIT    173070          OPREG = 172416          STRAM   173222
BIT04 = 000020          B2$     173750          INITLS  173014          OTBUF0  173366          WCR   = 172412
BIT05 = 000040          B3$     173776          IOBUF = 172416          OTBUF1  173402          ZEROBP  173420

. ABS.  174664     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM27=DK:PROM27
```

```
                                        .TITLE PROM NODE 28
                                        .IDENT  /V2.0/
000000                                  .ASECT
        173000                          .=173000


        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
        000002                          BIT01=  2
        000001                          BIT00=  1


173000  000005                  START:  RESET                           ;RESET BUS
173002  012700  000340                  MOV     #340,   R0              ;PRI=7
173006  106400                          MTPS    R0                      ;CPU PRI
173010  012706  160500                  MOV     #160500, SP             ;RESET STACK

                                        ;INITIALIZE LINE SWITCHES

173014  012737  010420  172416  INITLS: MOV     #4368., @#OPREG         ;MOD STAT
173022  012737  004414  172416          MOV     #4414,  @#OPREG         ;SET SWITCHES
173030  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173034  100375                          BPL     .-4                     ;LOOP UNTIL
173036  105037  172414                  CLRB    @#CSR
173042  012737  004540  172416          MOV     #4540,  @#OPREG         ;RESET SWITCHES
173050  105737  172414                  TSTB    @#CSR                   ;GOOD WD
173054  100375                          BPL     .-4                     ;LOOP UNTIL
173056  105037  172414                  CLRB    @#CSR
173062  013703  174652                  MOV     @#STATUS, R3            ;LC CATION OF STATUS
173066  005005                          CLR     R5                      ;CLEAR ERROR COUNT
```

```
 1                                                  ;INITIALIZE LIU
 2
 3 173070  005004                        INIT:   CLR     R4                      ;CLEAR LOOP COUNT
 4 173072  012737  010402  172416               MOV     #4354., @#OPREG         ;LDADR
 5 173100  012737  004400  172416               MOV     #2304., @#OPREG         ;ADDRESS=0
 6 173106  105737  172414                       TSTB    @#CSR                   ;GOOD WD
 7 173112  100375                              BPL     .-4                     ;NO RETRY
 8 173114  105037  172414                       CLRB    @#CSR
 9 173120  012737  010401  172416               MOV     #4353., @#OPREG         ;SEL ACRAM
10 173126  012700  000400                       MOV     #256., R0               ;COUNTER
11 173132  012737  004407  172416  1$:          MOV     #2311., @#OPREG         ;WRITE A NULL
12 173140  105737  172414                       TSTB    @#CSR                   ;GOOD WD
13 173144  100375                              BPL     .-4                     ;NO LOOP UNTIL READY
14 173146  105037  172414                       CLRB    @#CSR
15 173152  077011                              SOB     R0,     1$              ;LOOP UNTIL ZERO
16 173154  012700  000400                       MOV     #256.,  R0              ;COUNTER
17 173160  012737  001400  172416  2$:          MOV     #768.,  @#OPREG         ;RD
18 173166  105737  172414                       TSTB    @#CSR                   ;GOOD RD
19 173172  100375                              BPL     .-4                     ;NO RETRY
20 173174  105037  172414                       CLRB    @#CSR
21 173200  013701  172416                       MOV     @#IOBUF, R1             ;FETCH DATA
22 173204  042701  177760                       BIC     #177760, R1             ;CLEAR BITS
23 173210  022701  000007                       CMP     #7,      R1             ;DATA=NULL
24 173214  001401                              BEQ     3$                      ;YES
25 173216  005204                              INC     R4                      ;NO REPORT IT
26 173220  077021                        3$:   SOB     R0,     2$              ;LOOP UNTIL ZERO
27 173222  012737  010402  172416  STRAM:  MOV     #4354., @#OPREG         ;LDCTADR
28 173230  013737  174662  172416               MOV     @#NODE, @#OPREG         ;SEND ADDRESS
29 173236  105737  172414                       TSTB    @#CSR                   ;GOOD WRITE
30 173242  100375                              BPL     .-4                     ;LOOP UNTIL
31 173244  105037  172414                       CLRB    @#CSR
32 173250  012737  010401  172416               MOV     #4353., @#OPREG         ;SEL ACRAM
33 173256  012737  004404  172416               MOV     #4404,  @#OPREG         ;LOAD A 4
34 173264  105737  172414                       TSTB    @#CSR                   ;GOOD WRITE
35 173270  100375                              BPL     .-4                     ;NO LOOP UNTIL
36 173272  105037  172414                       CLRB    @#CSR
37 173276  012701  010410                INBUF0: MOV     #10410, R1              ;RDBUFADR CMD
38 173302  012700  010440                       MOV     #10440, R0              ;SEL INBUF0
39 173306  004767  000106                       JSR     PC,     ZEROBP          ;SET POINTER=0
40 173312  012737  001400  172416               MOV     #1400,  @#OPREG         ;FALSE READ DATA
41 173320  105737  172414                       TSTB    @#CSR                   ;GOOD READ
42 173324  100375                              BPL     .-4                     ;NO RETRY
43 173326  105037  172414                       CLRB    @#CSR                   ;CLEAR DONE BIT
44 173332  012701  010610                INBUF1: MOV     #10610, R1              ;RDBUFADR COMMAND
45 173336  012700  010640                       MOV     #10640, R0              ;SEL INBUF1 COMMAND
46 173342  004767  000052                       JSR     PC,     ZEROBP          ;POINTER=0
47 173346  012737  001400  172416               MOV     #1400,  @#OPREG         ;FALSE READ DATA
48 173354  105737  172414                       TSTB    @#CSR                   ;GOOD RD
49 173360  100375                              BPL     .-4                     ;NO RETRY
50 173362  105037  172414                       CLRB    @#CSR                   ;CLEAR DONE BIT
51 173366  012701  010510                OTBUF0: MOV     #10510, R1              ;RDBUFADR COMMAND
52 173372  012700  010540                       MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
53 173376  004767  000016                       JSR     PC,     ZEROBP
54 173402  012701  010710                OTBUF1: MOV     #10710, R1              ;RDBUFADR COMMAND
55 173406  012700  010740                       MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
56 173412  004767  000002                       JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
57 173416  000462                              BR      STAT                    ;GO CLEAR STATUS
```

```
 1                                          ;INITIALIZE BUFFERS & STATUS
 2
 3 173420  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
 4 173424  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
 5 173432  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 6 173436  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
 7 173440  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 8 173444  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
 9 173450  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
10 173454  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
11 173460  022702  000000          4$:     CMP     #0,     R2              ;POINTER=0
12 173464  001412                          BEQ     5$
13 173466  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
14 173474  105737  172414                  TSTB    @#CSR                   ;GOOD RD
15 173500  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
16 173502  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
17 173506  005302                          DEC     R2                      ;POINTER-1
18 173510  000763                          BR      4$
19 173512  010137  172416          5$:     MOV     R1,     @#OPREG         ;RDBUFADR
20 173516  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
21 173524  105737  172414                  TSTB    @#CSR                   ;GOOD RD
22 173530  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
23 173532  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
24 173536  013702  172416                  MOV     @#IOBUF, R2             ;FETCH NEW POINTER
25 173542  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
26 173546  022702  000000                  CMP     #0,     R2              ;POINTER=0
27 173552  001401                          BEQ     6$
28 173554  005204                          INC     R4                      ;REPORT ERROR
29 173556  010037  172416          6$:     MOV     R0,     @#OPREG         ;SEL BUFFER
30 173562  000207                          RTS     PC                      ;RETURN
31 173564  012737  010400  172416  STAT:   MOV     #4352., @#OPREG         ;WCR:RS
32 173572  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
33 173600  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
34 173606  013700  172416                  MOV     @#OPREG, R0             ;FETCH STATUS
35 173612  042700  177400                  BIC     #177400, R0             ;CLEAR MST BYTE
36 173616  022700  000000                  CMP     #0,     R0              ;STATUS=0 ?
37 173622  001401                          BEQ     7$
38 173624  005204                          INC     R4                      ;STATUS BAD REPORT
39 173626  022704  000000          7$:     CMP     #0,     R4              ;WERE THERE ERRORS
40 173632  001406                          BEQ     8$                      ;NO GOOD INIT
41 173634  005205                          INC     R5                      ;ERROR +1
42 173636  022705  000005                  CMP     #5,     R5              ;ERROR LIMIT YET
43 173642  001406                          BEQ     9$                      ;YES RETURN A 1(ERROR)
44 173644  000167  177220                  JMP     INIT                    ;NO RESTART VINIT
45 173650  012713  000001          8$:     MOV     #1,     (R3)            ;GOOD INIT.
46 173654  000167  000004                  JMP     SRTLDR
47 173660  012713  177777          9$:     MOV     #-1,    (R3)            ;LIU BAD INIT.
48                                         .ENABLE LSB
```

```
                                        ;LOADER START
                                        
173664  005077  000764        SRTLDR: CLR     @MEMPT                  ;MEMORY POINTER
173670  005077  000762                CLR     @PKCNT                  ;PACKET COUNTER
173674  005277  000756                INC     @PKCNT                  ;START'S AT 1
173700  105037  172414                CLRB    @#CSR                   ;CLEAR CSR
173704  005737  172414        LIO:    TST     @#CSR                   ;REQUEST YET
173710  100375                      BPL     .-4                     ;NO LOOP ON LIO
173712  013700  174650                MOV     @#DATA,  R0             ;ADDRESS OF DATA
173716  005010                      CLR     (R0)                    ;CLEAR DATA
173720  012706  160500                MOV     #160500,  SP            ;RESET SP
173724  012737  010400  172416        MOV     #4352., @#OPREG         ;WCR : RS(0)
173732  012737  002400  172416        MOV     #1280., @#OPREG         ;RD
173740  013701  172416                MOV     @#IOBUF, R1
173744  042701  177400                BIC     #177400, R1             ;CLEAR UNUSED BITS
                                        
                                        ;BUFFER OVERFLOW
                                        
173750  132701  000004        B2$:    BITB    #BIT02, R1              ;OV-FL
173754  001410                      BEQ     B3$
173756  012702  010410                MOV     #4360., R2              ;COM=SEL INBUF 0
173762  012703  010440                MOV     #4384., R3              ;COM=RDBUFPNT
173766  004767  000144                JSR     PC, EMBF                ;EMPTY BUFFER
173772  004767  000530                JSR     PC, NAK                 ;SEND NAK
173776  132701  000010        B3$:    BITB    #BIT03, R1              ;OV-FL
174002  001410                      BEQ     B01$
174004  012702  010610                MOV     #4488., R2              ;COM=SEL INBUF 1
174010  012703  010640                MOV     #4512., R3              ;COM=RDBUFPNT
174014  004767  000116                JSR     PC, EMBF
174020  004767  000502                JSR     PC, NAK                 ;SEND NAK
                                        
                                        ;BUFFERS FULL
                                        
174024  132701  000001        B01$:   BITB    #BIT00, R1              ;BUFFER 0 FULL
174030  001410                      BEQ     2$                      ;NO CK 1
174032  012702  010410        1$:     MOV     #4360., R2              ;YES, COM=RDBUFADR
174036  012703  010440                MOV     #4384., R3              ;COM=SEL INBUF0
174042  012704  000001                MOV     #BIT00, R4              ;WHICH CRC BIT
174046  004767  000026                JSR     PC,     PACK            ;GO FIND PACKET
174052  132701  000002        2$:     BITB    #BIT01, R1              ;BUFFER 1 FULL
174056  001712                      BEQ     LIO                     ;NO LOOP BACK
174060  012702  010610                MOV     #4488., R2              ;YES,COM=RDBUFADR
174064  012703  010640                MOV     #4512., R3              ;COM=SEL INBUF1
174070  012704  000002                MOV     #BIT01, R4              ;WHICH CRC BIT
174074  004767  000000                JSR     PC,     PACK            ;GO FIND PACKET
```

```
 1                                          ;UNLOAD BUFFERS
 2
 3 174100  005010                  PACK:    CLR     (R0)                ;CLEAR STATUS
 4 174102  004767  000030                   JSR     PC,     EMBF        ;EMPTY BUFFER
 5 174106  005710                           TST     (R0)                ;GOOD UNLOAD
 6 174110  100003                           BPL     4$                  ;YES
 7 174112  004767  000410          3$:      JSR     PC,     NAK         ;NO NAK TO REPLACE
 8 174116  000207                           RTS     PC                  ;RETURN
 9 174120  004767  000226          4$:      JSR     PC,     LOAD        ;LOAD + CONTROL PARSE
10 174124  005710                           TST     (R0)                ;CHECK RESPONSE
11 174126  100771                           BMI     3$                  ;BAD NAK
12 174130  004767  000364                   JSR     PC,     ACK         ;ACK IT AND CONTINUE
13 174134  000207                           RTS     PC
14 174136  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG     ;RS (CRC)
15 174144  012737  002400  172416           MOV     #1280., @#OPREG     ;RS CMD
16 174152  013705  172416                   MOV     @#IOBUF, R5         ;FETCH STATUS
17 174156  130405                           BITB    R4, R5              ;CRC OK
18 174160  001002                           BNE     5$                  ;YES CONTINUE
19 174162  012710  177777                   MOV     #-1,    (R0)        ;FLAG BAD CRC
20 174166  012737  160000  172410  5$:      MOV     #160000, @#BAR      ;ADDRESS TO WRITE
21 174174  010237  172416                   MOV     R2,     @#OPREG     ;RDBUFADR
22 174200  012737  001400  172416           MOV     #768.,  @#OPREG     ;READ POINTER
23 174206  105737  172414                   TSTB    @#CSR               ;GOOD RD
24 174212  100375                           BPL     .-4                 ;NO LOOP UNTIL READY
25 174214  105037  172414                   CLRB    @#CSR               ;CLEAR DONE BIT
26 174220  013704  172416                   MOV     @#IOBUF, R4         ;READ POINTER
27 174224  042704  177400                   BIC     #177400, R4         ;CLEAR UNUSED BITS
28 174230  022704  000004                   CMP     #4,     R4          ;CONTROL PACKET
29 174234  001405                           BEQ     6$                  ;YES UNLOAD
30 174236  022704  000204                   CMP     #132.,  R4          ;MEMORY IMAGE UNLOAD
31 174242  001402                           BEQ     6$
32 174244  012710  177777                   MOV     #-1,    (R0)        ;BAD UNLOAD
33 174250  005404                  6$:      NEG     R4                  ;2'S COMP
34 174252  010437  172412                   MOV     R4,  @#WCR          ;BYTE COUNT
35 174256  010337  172416                   MOV     R3,     @#OPREG     ;SEL BUFFER
36 174262  012737  001400  172416           MOV     #768.,  @#OPREG     ;FALSE RD
37 174270  105737  172414                   TSTB    @#CSR               ;GOOD RD
38 174274  100375                           BPL     .-4                 ;NO LOOP UNTIL READY
39 174276  105037  172414                   CLRB    @#CSR
40 174302  012737  021600  172416           MOV     #8704., @#OPREG     ;FIRE DMA RD
41 174310  000240                           NOP                         ;DELAY
42 174312  105737  172414                   TSTB    @#CSR               ;GOOD DMA
43 174316  100402                           BMI     7$                  ;YES
44 174320  012710  177777                   MOV     #-1,  (R0)          ;BAD DMA XFER
45 174324  105037  172414          7$:      CLRB    @#CSR               ;CLEAR CSR
46 174330  012737  004400  172416           MOV     #2304., @#OPREG     ;RESET POINTER TO 255
47 174336  105737  172414                   TSTB    @#CSR               ;GOOD WD
48 174342  100375                           BPL     .-4                 ;NO RETRY
49 174344  105037  172414                   CLRB    @#CSR               ;CLEAR DONE BIT
50 174350  000207                  8$:      RTS     PC
51                                          .ENABLE LSB
```

```
 1                                          ;PACKET PARSE
 2
 3 174352  016702  000302          LOAD:   MOV     BUFFER, R2              ;BUFFER AREA
 4 174356  122712  000000                  CMPB    #0, (R2)                ;MEMORY OR CONTROL
 5 174362  001426                          BEQ     CMEQ                    ;CONTROL PACKET
 6 174364  016703  000266                  MOV     PKCNT,  R3              ;PACKET COUNTER
 7 174370  022713  000144                  CMP     #100.,  (R3)            ;LIMIT
 8 174374  001002                          BNE     1$                      ;NO CONTINUE
 9 174376  005013                          CLR     (R3)                    ;RESET POINTER
10 174400  005213                          INC     (R3)
11 174402  121312                  1$:     CMPB    (R3),   (R2)            ;NEW PACKET
12 174404  001401                          BEQ     2$
13 174406  000207                          RTS     PC                      ;RETURN
14 174410  062702  000002          2$:     ADD     #2,     R2              ;OFFSET HEADER
15 174414  005213                          INC     (R3)                    ;COUNT + 1
16 174416  017703  000232                  MOV     @MEMPT, R3
17 174422  012704  000100                  MOV     #64.,   R4              ;BYTE COUNT
18 174426  012223                  3$:     MOV     (R2)+, (R3)+            ;LOAD MEMORY
19 174430  077402                          SOB     R4,     3$              ;LOOP UNTIL ZERO
20 174432  010377  000216                  MOV     R3,     @MEMPT
21 174436  000207                          RTS     PC
22 174440  122762  000001  000001  CMEQ:   CMPB    #1,     1(R2)           ;IS IT A REPORT
23 174446  001004                          BNE     5$                      ;NO
24 174450  013703  174652                  MOV     @#STATUS, R3            ;LOCATION OF STATUS
25 174454  011310                          MOV     (R3),   (R0)            ;TRANSFER STATUS
26 174456  000207                          RTS     PC
27 174460  122762  000005  000001  5$:     CMPB    #5,     1(R2)           ;IS IT A START
28 174466  001005                          BNE     6$                      ;NO
29 174470  013700  000040                  MOV     @#40,   R0              ;FETCH START ADDRESS
30 174474  012706  001000                  MOV     #1000, SP               ;RESET USER STACK
31 174500  000110                          JMP     (R0)                    ;LEAVE PROM TO RAM
32 174502  122762  000006  000001  6$:     CMPB    #6,     1(R2)           ;WAS IT A LOAD COMMAND
33 174510  001002                          BNE     7$                      ;NO
34 174512  000167  177146                  JMP     SRTLDR                  ;YES GOTO LOAD START
35 174516  000207                  7$:     RTS     PC
36
```

```
                                                ;ACK/NAK ROUTINE
                                                
174520  012705  174630          ACK:    MOV     #ACKPK,   R5                ;ADDRESS OF ACK
174524  000402                          BR      8$
174526  012705  174640          NAK:    MOV     #NAKPK,   R5                ;ADDRESS OF NAK
174532  012737  010540  172416  8$:     MOV     #4448., @#OPREG             ;SEL OUTBF0
174540  012504                          MOV     (R5)+,  R4                  ;FETCH HEADER
174542  067704  000110                  ADD     @PKCNT, R4                  ;ADD PACKET COUNT
174546  010437  172416                  MOV     R4,     @#OPREG             ;WRITE IT
174552  105737  172414                  TSTB    @#CSR                       ;GOOD WD
174556  100375                          BPL     .-4                         ;NO LOOP
174560  012704  000003                  MOV     #3,     R4                  ;BYTE COUNT
174564  012537  172416          9$:     MOV     (R5)+, @#OPREG              ;WD
174570  105737  172414                  TSTB    @#CSR                       ;GOOD WD
174574  100375                          BPL     .-4                         ;NO LOOP
174576  077406                          SOB     R4,     9$                  ;YES LOOP TIL 0
174600  012737  010420  172416          MOV     #4368., @#OPREG             ;MODSTAT
174606  012737  004421  172416          MOV     #4421,  @#OPREG             ;SET BITS
174614  105737  172414                  TSTB    @#CSR                       ;GOOD WD
174620  100375                          BPL     .-4                         ;LOOP UNTIL
174622  105037  172414                  CLRB    @#CSR
174626  000207                          RTS     PC

174630  004400                  ACKPK:  .WORD   4400
174632  004402                          .WORD   4402
174634  004403                          .WORD   4403
174636  004405                          .WORD   4405
174640  004400                  NAKPK:  .WORD   4400
174642  004403                          .WORD   4403
174644  004403                          .WORD   4403
174646  004405                          .WORD   4405
174650  160400                  DATA:   .WORD   160400
174652  160402                  STATUS: .WORD   160402
174654  160404                  MEMPT:  .WORD   160404
174656  160406                  PKCNT:  .WORD   160406
174660  160000                  BUFFER: .WORD   160000
174662  004402                  NODE:   .WORD   4402

        173000                          .END    START
```

```
PROM NODE 28   MACRO V03.01 27-MAR-79 18:59:38 PAGE 7-1
SYMBOL TABLE

ACK     174520          BIT06 = 000100          CMEQ    174440          LIO     173704          PACK    174100
ACKPK   174630          BIT07 = 000200          CSR   = 172414          LOAD    174352          PKCNT   174656
BAR   = 172410          BIT08 = 000400          DATA    174650          MEMPT   174654          SRTLDR  173664
BIT00 = 000001          BIT14 = 040000          EMBF    174136          NAK     174526          START   173000
BIT01 = 000002          BIT15 = 100000          INBUF0  173276          NAKPK   174640          STAT    173564
BIT02 = 000004          BUFFER  174660          INBUF1  173332          NODE    174662          STATUS  174652
BIT03 = 000010          B01$    174024          INIT    173070          OPREG = 172416          STRAM   173222
BIT04 = 000020          B2$     173750          INITLS  173014          OTBUF0  173366          WCR   = 172412
BIT05 = 000040          B3$     173776          IOBUF = 172416          OTBUF1  173402          ZEROBP  173420

. ABS.  174664     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:PROM28=DK:PROM28
```

```
                                        .TITLE  SIG LOADING PROM
                                        .IDENT  /V1.0/
000000                                  .ASECT
        173000                          .=173000


        000000                          L.CKSM= %0
        000000                          R0=     %0
        000001                          L.ADR=  %1
        000001                          R1=     %1
        000002                          L.BC=   %2
        000002                          R2=     %2
        000003                          L.BYT=  %3
        000003                          R3=     %3
        000004                          R4=     %4
        000005                          L.PTR=  %5
        000005                          R5=     %5
        000006                          SP=     %6
        000007                          PC=     %7

        175610                          TKS=    175610
        175612                          TKB=    175612
        175614                          TPS=    175614
        175616                          TPB=    175616


173000  000005                  SIGLDR: RESET
173002  012700  000340                  MOV     #340,   R0              ;CPU PRI=7
173006  106400                          MTPS    R0
173010  012706  160500                  MOV     #160500, SP             ;RESET STACK
173014  012737  000122  175616  1$:     MOV     #'R,    @#TPB           ;SIGNAL READY
173022  105737  175610                  TSTB    @#TKS                   ;FEED BACK
173026  100372                          BPL     1$
173030  013700  175612                  MOV     @#TKB,  R0                      ;FETCH BYTE
173034  042700  177400                  BIC     #177400,R0              ;CLEAR MST
173040  022700  000114                  CMP     #'L,    R0              ;LOAD MODE
173044  001363                          BNE     1$                      ;NO STAY READY
173046  000005                          RESET                           ;RESET BUS FOR ABSLDR


```

```
 1
 2 173050  012716  175610          ABSL:   MOV     #TKS,(SP)       ;--> READER CSR
 3
 4 173054  011646                  L.LOAD: MOV     (SP),-(SP)      ;MAKE 2 COPIES FOR L.READ
 5 173056  012705  173170          L.LD1:  MOV     #L.READ,L.PTR   ;--> READ ROUTINE
 6 173062  005001                          CLR     L.ADR           ;CLEAR THE ROAD ADDRESS
 7 173064  005004                          CLR     R4              ;CLEAR RELOCATION
 8 173066  010446                  L.LD1B: MOV     R4,-(SP)        ;PICK UP THE CONTENT OF
 9                                                                 ;THE SOFTWARE SWITCH REGISTER
10 173070  006016                          ROR     @SP             ;CHECK RELOCATION FACTOR
11 173072  103402                          BCS     L.LD1C          ;JUMP IF SOME RELOCATION NEEDED
12 173074  005016                          CLR     @SP             ;GO DO LOAD
13 173076  000403                          BR      L.LD2           ;GO DO LOAD
14 173100  006316                  L.LD1C: ASL     @SP             ;CHECK FOR NON-ZERO
15 173102  001001                          BNE     L.LD2           ;JUMP IF LOAD ADDRESS SPECIFIED
16 173104  010116                          MOV     L.ADR,@SP       ;OTHERWISE CONTINUE LOADING FROM LAST LOAD
17
18                                         ;*  LOOK FOR THE BEGINNING OF A BLOCK
19
20 173106  005000                  L.LD2:  CLR     L.CKSM          ;INITIALIZE CHECKSUM
21 173110  004715                          JSR     PC,@L.PTR       ;READ A FRAME
22 173112  105303                          DECB    L.BYT           ;LOOP UNTIL +1 (START OF A BLOCK)
23 173114  001374                          BNE     L.LD2           ;LOOP UNTIL +1 IS FOUND
24 173116  004715                          JSR     PC,@L.PTR       ;READ ANOTHER FRAME
25
26                                         ;*
27                                         ;*  INPUT AND SAVE BYTE COUNT, IF BYTE COUNT IS IQUAL TO 6
28                                         ;*  GO TO PROCEED JUMP
29                                         ;*
30
31 173120  004767  000074                  JSR     PC,L.GWRD       ;GET FULL BYTE COUNT
32 173124  010402                          MOV     R4,L.BC         ;
33 173126  062702  177774                  ADD     #-4,L.BC        ;SUBTRACT 4 TO MAKE BYTE COUNT CORRECT
34 173132  022702  000002                  CMP     #2,L.BC         ;WAS BYTE COUNT EQUAL TO 6?
35 173136  001436                          BEQ     L.JMP           ;JUMP IF NO DATA (E.G. - JUMP BLOCK)
36 173140  004767  000054                  JSR     PC,L.GWRD       ;GET LOAD ADDRESS
37 173144  061604                          ADD     @SP,R4          ;GENERATE ACTUAL ADDRESS
38 173146  010401                          MOV     R4,L.ADR        ;AND PUT IT INTO THE PROPER CELL
39
40                                         ;*
41                                         ;*  READ IN REMAINDER OF DATA
42                                         ;*  IF THE LOADER HALTS AT L.BAD, A CHECKSUM ERROR
43                                         ;*  HAS OCCURED, R3 WILL CONTAIN THE EXPECTED CHECKSUM,
44                                         ;*  AND R0 WILL CONTAIN THE DEVIATION FROM THE EXPECTED
45                                         ;*  CHECKSUM.
46                                         ;*
47 173150  004715                  L.LD3:  JSR     PC,@L.PTR       ;READ A FRAME
48 173152  002004                          BGE     L.LD4           ;BRANCH IF MORE DATA REMAINS
49 173154  105700                          TSTB    L.CKSM          ;IF CHECKSUM IS
50 173156  001753                          BEQ     L.LD2           ;CORRECT, THEN CONTINUE
51 173160  000000                  L.BAD:  HALT                    ;CHECKSUM ERROR
52 173162  000751                          BR      L.LD2           ;PRESS CONTINUE TO IGNORE CHECKSUM
53 173164  110321                  L.LD4:  MOVB    L.BYT,(L.ADR)+  ;STORE 8 BITS AT A TIME
54 173166  000770                          BR      L.LD3           ;    THE RE-LOOP
55
```

```
 1                                          ;*
 2                                          ;*  INPUT A FRAME, DECREMENT BYTE COUNT, AND ACCUMULATE CHECKSUM
 3                                          ;*
 4 173170  016603  000006          L.READ: MOV     6(SP),L.BYT     ;DEVICE ADDRESS TO L.BYT
 5 173174  105213                          INCB    @L.BYT          ;SELECT READER
 6 173176  105713                  L.R1:   TSTB    @L.BYT          ;DONE? -
 7 173200  100376                          BPL     L.R1            ;NO
 8 173202  116303  000002                  MOVB    2(L.BYT),L.BYT  ;GET CHARACTER
 9 173206  042703  177400                  BIC     #177400,I.BYT   ;CLEAR GARBAGE BITS
10 173212  060300                          ADD     L.BYT,L.CKSM    ;ADD TO CHECKSUM
11 173214  005302                          DEC     L.BC            ;DECREMENT BYTE COUNT BY ONE
12 173216  000207                          RTS     PC
13
14
15                                          ;*
16                                          ;*  ASSEMBLE ONE FULL WORD OF DATA
17                                          ;*
18 173220  004715                  L.GWRD: JSR     PC,@L.PTR       ;GET ONE CHAR
19 173222  010304                          MOV     L.BYT,R4        ;SAVE R3 IN TEMP
20 173224  004715                          JSR     PC,@L.PTR       ;GET ANOTHER FRAME
21 173226  000303                          SWAB    L.BYT           ;PLACE A FRAME
22 173230  050304                          BIS     L.BYT,R4        ;ASSEMBLE INTO ONE WORD
23 173232  000207                          RTS     PC
24
25                                          ;*
26                                          ;*  CHECK CORRECTNESS OF JUMP ADDRESS
27                                          ;*  HALT IF ADDRESS IS ODD, JUMP TO PROGRAM IF ADDRESS IS EVEN
28                                          ;*
29 173234  004767  177760          L.JMP:  JSR     PC,L.GWRD       ;GET POSSIBLE TRANSFER ADDRESS
30 173240  004715                          JSR     PC,@L.PTR       ;GET CHECKSUM
31 173242  105700                          TSTB    L.CKSM          ;IF INCORRECT
32 173244  001345                          BNE     L.BAD           ;  GO TO CHECKSUM HALT ADDRESS
33 173246  006204                          ASR     R4              ;GET LOW ORDER BIT
34 173250  103002                          BCC     L.JMP1          ;SKIP IF ADDRESS IS EVEN
35 173252  000000                          HALT                    ;OTHERWISE HALT
36 173254  000704                          BR      L.ID1B          ;RETURN TO START OF LOADING LOOP
37 173256  006304                  L.JMP1: ASL     R4              ;RESTORE REGISTER
38 173260  061604                          ADD     @SP,R4
39 173262  000114                          JMP     @R4             ;JUMP TO USER
40
41
42         173000                           .END    SIGLDR
```

```
SIG LOADING PROM        MACRO V03.01 27-MAR-79 19:00:35 PAGE 3-1
SYMBOL TABLE

ABSL    173050          L.CKSM=%000000          L.LD1B  173066          L.LOAD  173054          TKB    = 175612
L.ADR =%000001          L.GWRD  173220          L.LD1C  173100          L.PTR =%000005          TKS    = 175610
L.BAD   173160          L.JMP   173234          L.ID2   173106          L.READ  173170          TPB    = 175616
L.BC  =%000002          L.JMP1  173256          L.ID3   173150          L.R1    173176          TPS    = 175614
L.BYT =%000003          L.LD1   173056          L.LD4   173164          SIGLDR  173000

. ABS.  173264      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  288 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:SIGLDR=DK:SIGLDR
```

4.0 MODIFICATION TO ESMD SOFTWARE

STACK DIRECTORY FRIDAY 05/25/79 12:31:45

FILE-NAME	USER CODE	CREATE DATE	CREATE TIME	LAST ACCESS DATE	LAST ACCESS TIME	RECORD SIZE	BLOCK SIZE	RECORD COUNT	PAGE SIZE	FILE TYPE	DISK ADDR	TOTAL SECTORS
AMU.	WAYN	08/12/77	15:47	10/30/79	13:38	45	45	8	8	2	1396	8
TEST-48HOUR.	MEL	04/27/78	19:46	04/05/79	08:59	45	45	4	4	2	1007	4
ALGOL-TABLE.	BOSS	03/11/75	01:02	05/25/79	09:02	45	45	175	175	0	325	175
DIRECTORY.	BOSS	03/11/75	01:02	05/25/79	12:31	45	45	135	135	0	500	135
ESMLDR1.	MEL	03/18/78	15:56	03/22/78	17:53	45	45	16	16	2	25648	16
SUN-RISE.	SYST	03/17/77	09:53	05/25/79	08:12	45	45	20	20	2	1107	20
TCC14.	MEL	09/07/78	17:57	04/23/79	26:54	45	45	20	20	4	6519	20
CARD-LIST.	SYST	01/28/77	12:11	04/23/79	23:50	45	45	21	21	2	1135	21
COLD-START.	BOSS	03/11/75	01:02	07/13/77	16:39	45	45	7	7	2	1000	7
OBJ-WTDR.	MEL	05/11/78	14:20	05/11/78	14:20	45	180	2	4	0	19694	4
SYSTEM.	SYST	03/11/75	01:02	05/25/79	12:28	45	0	48	48	1	1013	48
SYSTEM/DIR-LIST.	SYST	09/23/77	08:36	05/25/79	12:31	45	45	17	17	2	4302	17
SYSTEM/LOAD.	SYST	12/14/76	09:01	07/17/78	09:11	45	45	16	16	2	1083	16
SYSTEM/DISK-MAP.	BOSS	03/11/75	01:02	05/25/79	12:30	45	45	199	199	0	800	199
SYSTEM/MCP.	BOSS	03/11/75	01:02	03/17/78	14:54	45	45	198	198	4	2	198
SYSTEM/ALGOL.	BOSS	03/11/75	01:02	02/16/78	10:45	45	45	125	125	2	200	125
SYSTEM/BOOTSTRAP.	BOSS	03/11/75	01:02	03/11/75	01:03	45	45	5	5	0	635	5
SYSTEM/STACK.	BOSS	03/11/75	01:02	04/15/78	12:16	45	45	160	160	0	640	160
SYSTEM/MCP-SEG.	BOSS	03/11/75	01:02	10/30/79	13:38	45	45	1	1	0	999	1
SYSTEM/GROUP.	SYST	03/11/75	01:02	06/08/78	10:20	45	45	6	6	2	1160	6
SYSTEM/TRANSLANG.	SYST	08/09/77	15:40	03/17/78	13:57	45	45	98	98	2	9357	98
SYSTEM/DISC-PRINT.	SYST	02/03/76	10:03	03/27/78	14:57	45	45	8	8	2	2048	8
SYSTEM/SEQ-COMPARE.	RON	01/18/77	16:07	06/15/77	08:27	45	45	8	8	2	1099	8
SYSTEM/MAP-OUT.	RON	03/11/75	01:03	06/17/77	13:59	45	45	6	6	2	2056	6
SYSTEM/DSIM.	SYST	07/12/77	12:07	07/12/77	12:12	45	45	94	94	2	9785	94
SYSTEM/DUMP.	SYST	12/14/76	09:00	07/17/78	09:09	45	45	16	16	2	3912	16
SYSTEM/MT-PRINT.	SYST	11/22/76	13:36	05/11/78	09:35	45	45	10	10	2	3928	10
SYSTEM/ASSIGN.	WAYN	01/16/78	09:31	10/30/79	13:38	20	180	45	20	3	1375	20
SYSTEM/CURRENTSTATU.	MANG	11/22/77	12:27	02/23/78	15:05	969	969	1	22	0	3419	22
SYSTEM/DECK-COMPARE.	RON	02/24/77	08:48	06/15/77	08:41	45	45	8	8	2	9769	8
SYSTEM/DISK-COMPARE.	RON	01/18/77	10:34	02/08/78	09:41	45	45	11	11	2	7058	11
SYSTEM/MDMPL.	JIM	10/14/76	13:13	06/15/77	08:32	45	45	71	71	2	4417	71
SYSTEM/COPY-TOP-BOT.	SYST	05/06/77	08:00	05/25/79	10:05	45	45	10	10	2	7233	10
OBJ-WTDS.	MEL	05/11/78	14:21	05/11/78	14:21	45	180	2	4	0	1166	4
SOURCE.	SYST	05/19/76	15:54	05/25/79	12:28	45	45	48	48	1	1272	48
SOURCE/WTDR.	MEL	01/10/78	13:08	05/11/78	14:19	20	1800	176	80	3	11601	80
SOURCE/WHO.	WAYN	05/03/77	10:04	06/15/77	08:28	20	180	58	28	3	1440	28
SOURCE/WTDS.	MEL	01/10/78	13:06	05/11/78	14:20	20	1800	176	80	3	6094	80
SOURCE/CRT.	SYST	05/12/76	10:57	12/13/77	08:37	20	180	14	8	3	1172	8
SOURCE/NODE-CNTRL.	MEL	09/07/78	17:31	05/25/79	09:02	20	1800	949	160	3	17269	480
SOURCE/BDS-STACK.	MEL	05/11/78	14:30	05/25/79	12:16	20	1800	4633	160	3	10126	2080
SOURCE/HOST-TEST.	MEL	03/18/78	12:42	03/18/78	12:42	20	180	30	16	3	1772	16
SOURCE/NODAL-PATCH.	MEL	04/23/79	21:36	05/25/79	09:02	20	180	85	40	3	1727	40
SOURCE/STACK-PATCH.	MEL	07/19/79	14:11	05/25/79	12:16	20	180	81	36	3	10700	36
SOURCE/LOADER-PATCH.	MEL	04/04/79	18:52	05/25/79	08:09	20	180	5	4	3	6803	4
SOURCE/TEST-48HOUR.	MEL	04/27/78	19:39	06/28/78	11:05	20	180	34	16	3	6807	16
SOURCE/TAPE-OP.	DAN	07/13/77	13:17	06/28/78	10:28	20	180	813	160	3	10777	480
SOURCE/USRLNG-PATCH.	DAN	04/23/79	26:35	05/25/79	08:13	20	180	11	8	3	4649	8
SOURCE/DUMMY.	MEL	04/23/79	20:09	04/23/79	20:12	20	180	38	20	3	10654	20
SOURCE/LIST.	MEL	04/23/79	20:50	04/23/79	20:52	20	180	84	80	3	7069	80
SOURCE/STACK-LIST.	MEL	04/23/79	23:49	04/23/79	24:23	20	180	79	80	3	7149	80
SOURCE/TABLE.	MEL	03/28/78	14:17	03/28/78	14:46	20	180	19	12	3	2193	12
SOURCE/FILE-FIA.	MEL	12/05/77	16:15	05/25/79	15:17	20	180	39	20	3	5224	20
SOURCE/CRT-READ.	MEL	02/17/78	54:10	04/23/79	24:37	20	180	18	8	3	17911	8
SOLC14.	MEL	04/27/78	09:30	04/27/78	09:30	45	45	20	20	4	1188	20
TRANSLANG-TA.	SYST	07/13/77	13:29	03/17/78	15:46	45	45	91	91	0	11719	91
MUMPL-TABLE.	JIM	03/08/76	09:43	06/15/77	08:31	45	45	104	104	0	4103	104
OPERATOR.	SYST	01/29/76	19:00	05/25/79	12:28	45	45	16	16	1	4207	16
OPERATOR/DEFINES.	MEL	01/19/78	15:59	06/28/78	10:28	20	180	15	8	3	6421	8
TCC15.	MEL	09/07/78	17:59	09/07/78	17:59	45	45	20	20	4	6539	20
BDSIM.	MEL	08/31/77	13:06	02/25/78	12:56	45	45	99	99	2	8868	99
CRC-CALC.	CJT	09/21/77	17:11	04/13/78	14:14	20	180	27	12	3	4081	12
GAI12.	MEL	04/05/79	11:30	05/25/79	08:[illegible]	45	45	25	25	4	5774	25
ESMU.	DAN	01/06/78	11:16	05/25/79	12:28	45	45	16	16	1	1252	16
ESMU/MEM.	DAN	02/09/78	14:55	05/11/78	14:08	20	1800	233	120	3	7956	120
ESMU/LPFT-FOR.	DAN	01/06/78	11:20	01/23/78	18:57	20	180	204	92	3	2514	92
ESMU/FTMSG-DAT.	DAN	01/06/78	11:21	01/10/79	10:08	20	180	26	12	3	1225	12
ESMU/OBJ-MEM.	DAN	05/11/78	14:09	04/23/79	19:51	45	180	2	4	0	1268	4
ESMU/LPFT-OBJ.	DAN	01/06/78	11:22	01/23/78	18:56	45	45	15	15	2	1571	15
ESMU/LPFT.	DAN	05/03/78	43:36	06/28/78	10:27	20	1800	444	160	3	22591	320
ESMU/BDS-STACK.	MEL	05/11/78	14:36	05/25/79	08:15	45	180	31	32	0	1898	32
ESMU/PDP-BDS-STAC.	MEL	05/11/78	15:22	05/25/79	08:08	45	180	29	32	0	1829	32
ESMU/B776-STACK.	SYST	04/12/78	13:37	04/15/78	12:16	180	180	11	44	6	9902	44
ESMU/ESMLDR.	MEL	05/05/78	13:29	05/25/79	08:09	20	1800	404	160	3	7338	320
ESMU/SDLC-BDS-STA.	MEL	05/11/78	15:31	05/25/79	08:07	45	180	29	32	0	1695	32

ESMD/GAT13-BDS-SI.	MEL	05/25/79	12:23	05/25/79	12:23	45	180	31	32	0	1324	32
ESMD/GAT13-ESMD-S.	MEL	05/25/79	10:33	05/25/79	10:39	45	180	31	32	0	2735	32
GAT13.	MEL	04/23/79	21:39	05/25/79	08:12	45	45	26	26	4	10674	26
SDL15.	MEL	09/07/78	18:02	05/25/79	08:07	45	45	20	20	4	6559	20
USRLNG.	CMT	04/23/79	26:38	05/25/79	08:42	45	45	44	44	2	2398	44
HST16.	MEL	04/05/79	11:36	05/25/79	08:16	45	45	23	23	4	4626	23
EE.	RON	03/30/77	10:32	12/01/77	14:13	45	45	46	46	2	5977	46
RCMAIN.	TRW	08/19/77	08:17	05/25/79	12:23	45	45	82	82	2	4667	82
SDL14.	MEL	09/07/78	17:40	05/25/79	08:07	45	45	20	20	4	17473	20
MASTER.	WAYN	01/03/78	14:57	02/23/78	15:05	45	45	103	103	2	1468	103
RC.	TRW	05/25/76	10:13	05/25/79	12:15	45	45	6	6	2	6328	6
IRW.	TRW	04/24/78	08:41	05/25/79	12:29	45	45	16	16	1	29166	16
IRW/SOURCE.	TRW	04/24/78	08:42	05/25/79	12:29	45	45	16	16	1	1586	16
IRW/SOURCE/TCCF-RCV.	TRW	04/27/78	17:15	04/27/78	17:15	20	180	49	24	3	1788	24
IRW/SOURCE/TCCF-SEND.	TRW	04/24/78	11:22	04/25/78	15:40	20	180	53	24	3	2817	24
LOGOFF.	WAYN	12/09/77	08:34	04/05/79	18:49	45	45	4	4	2	2883	4
TAPE-OP.	DAN	03/17/78	15:47	07/17/78	09:12	180	180	2	8	6	12729	8
LPPT.	DAN	05/04/78	21:37	01/10/79	10:08	45	45	21	21	2	1962	21
ULU.	MEL	04/19/78	20:07	05/25/79	12:29	45	45	32	32	1	8146	32
ULU/TCC14.	MEL	05/11/78	12:35	05/11/78	12:35	45	45	20	20	4	1861	20
ULU/TCC15.	MEL	05/11/78	12:39	05/11/78	12:39	45	45	20	20	4	1930	20
ULU/SDL14.	MEL	06/08/78	12:13	09/07/78	16:55	45	45	20	20	4	4516	20
ULU/SDL15.	MEL	06/08/78	12:16	09/07/78	17:46	45	45	20	20	4	4536	20
ULU/HST16.	MEL	05/11/78	12:24	09/07/78	17:47	45	45	23	23	4	1618	23
ULU/GAT17.	MEL	05/11/78	12:44	09/07/78	17:46	45	45	19	19	4	2003	19
ULU/CRT18.	MEL	05/11/78	12:47	09/07/78	17:46	45	45	23	23	4	2170	23
ULU/GAT12.	MEL	05/11/78	12:27	09/07/78	16:55	45	45	25	25	4	2136	25
ULU/ESMLDR.	MEL	03/24/78	16:22	05/05/78	11:50	45	45	17	17	2	2078	17
ULU/GAT13.	MEL	04/05/79	11:24	04/23/79	19:32	45	45	26	26	4	5748	26
ULU/SEC19.	MEL	05/11/78	13:00	09/07/78	17:47	45	45	26	26	4	25839	26
ULU/ESMD-GAT13-B.	MEL	04/05/79	09:41	04/23/79	19:58	45	180	31	32	0	5676	32
ULU/USRLNG.	CMT	04/05/79	09:17	04/23/79	23:10	45	45	43	43	2	3441	43
SEQLIST.	WAYN	11/09/77	11:02	07/18/79	15:06	45	45	15	15	2	1603	15
JUNK.	DAN	04/27/78	21:09	04/27/78	22:49	20	180	4	80	3	4749	80
GAT17.	MEL	09/07/78	18:06	05/25/79	08:07	45	45	19	19	4	6602	19
CHECK-ATTACH.	WAYN	08/05/77	10:09	05/08/78	14:02	45	45	6	6	2	11370	6
DIAGLDR.	DAN	03/24/78	17:44	04/23/79	24:39	45	45	11	11	2	2036	11
CRT18.	MEL	09/07/78	18:08	05/25/79	08:11	45	45	23	23	4	6621	23
8776-SEC.	DAN	04/27/78	18:07	06/28/78	11:00	20	180	17	80	3	4884	80
OBJ-8776-SEC.	DAN	05/05/78	14:18	01/10/79	09:05	45	45	3	3	2	1069	3
LENFILE.	LEN	06/28/78	09:58	07/17/78	10:21	20	180	12	8	3	4556	8
END.	WAYN	09/15/77	08:36	05/25/79	09:27	45	45	7	7	2	22359	7
CRC.	CJT	10/21/77	12:41	04/13/78	14:06	45	45	4	4	2	1073	4
PARSE-OP.	SYST	07/20/77	12:49	05/25/79	12:16	180	180	1	4	6	1127	4
MSG.	DAN	05/03/78	44:59	05/03/78	45:15	20	180	213	80	3	8338	160
SLAVE.	WAYN	10/18/77	26:18	05/08/78	14:02	45	45	10	10	2	1361	10
BDS-TABLE.	MEL	08/31/77	13:07	02/25/78	12:57	256	256	1	6	0	1412	6
SEC19.	MEL	09/07/78	18:11	05/25/79	08:08	45	45	26	26	4	6644	26
LEARN.	LEN	06/08/78	10:24	09/08/78	09:04	20	180	24	12	3	4504	12
USER.	CMT	11/11/77	12:41	05/25/79	12:29	45	45	16	16	1	1660	16
USER/P0000.	CMT	04/23/79	26:07	05/25/79	08:13	20	1800	1012	160	3	7606	480
USER/INFO.	CMT	11/11/77	16:19	05/25/79	08:56	20	180	135	80	3	4976	80
USER/CKTD.	CMT	11/21/77	09:26	04/26/79	64:57	20	180	7	4	3	1434	4
USER/TNKD.	CMT	11/21/77	09:30	04/05/79	14:56	20	180	10	4	3	2813	8
USER/CDIR.	CMT	11/18/77	16:59	04/26/79	64:57	20	180	1	4	3	1408	4
USER/TDIR.	CMT	11/18/77	17:00	04/26/79	58:14	20	180	2	4	3	25835	4
ESMLDR.	MEL	04/04/79	18:53	05/25/79	08:15	45	45	18	18	2	2250	18
MSG.	SYST	03/23/78	12:55	05/25/79	12:16	45	45	53	53	2	24227	53
LOOP.	TRW	06/19/77	12:43	05/25/79	12:29	45	45	48	48	1	9287	48
LOOP/UTILITY.	TRW	09/19/77	17:30	10/30/79	13:38	45	45	41	41	2	9455	41
LOOP/SEND-ME.	RON	05/20/77	15:00	05/25/79	08:56	45	45	7	7	2	1418	7
LOOP/BDS-LOADER.	TRW	01/31/78	11:46	03/17/78	08:14	45	45	7	7	2	1688	7
DEVICE.	WAYN	09/19/77	16:46	12/01/77	09:28	45	45	15	15	2	4287	15
8721-TAPE-OP.	TRW	10/24/77	15:51	03/17/78	08:12	180	180	2	8	6	1217	8
USCRT.	DAN	05/10/78	09:53	05/10/78	10:48	20	180	16	8	3	1237	8
USRLN.	CMT	06/02/77	13:49	05/25/79	12:29	45	45	16	16	1	7940	16
USRLN/MSG.	CMT	11/21/77	10:08	05/25/79	08:42	20	1800	181	80	3	7526	160
MSG-TABLE.	SYST	03/18/78	16:36	05/25/79	12:16	45	45	71	71	0	24759	71
OBJ-DSCRT.	DAN	05/10/78	10:31	05/10/78	10:48	45	45	3	3	2	1180	3
SLY-NANO.	SYST	07/13/77	13:30	07/13/77	13:39	1512	1512	1	34	0	5889	34
SLY-NANO.	SYST	02/03/77	14:38	03/17/78	15:46	3072	3072	1	69	0	9534	69
OBJ-DSCRT16.	DAN	05/10/78	10:48	05/10/78	10:50	45	45	3	3	2	1208	3
MEL.	MEL	01/27/78	13:53	05/25/79	12:29	45	45	16	16	1	1882	16
MEL/HOST-TEST.	MEL	03/16/78	12:43	04/25/78	11:07	45	45	4	4	2	1320	4
MEL/FILE-FIX.	MEL	01/26/78	16:46	05/09/78	14:08	45	45	5	5	2	1648	5
MEL/CRT-READ.	MEL	02/17/78	60:53	04/25/78	10:58	45	45	3	3	2	17751	3
MEL/TEST-48HOUR.	MEL	04/21/78	12:10	04/21/78	12:10	45	45	4	4	2	23445	4
MEL/BDS-LOADER.	MEL	02/16/78	16:06	04/13/78	11:34	45	45	4	4	2	3299	4
JUNK1.	DAN	05/11/78	13:03	05/11/78	13:03	20	180	3	4	3	1245	4
EE-TABLE.	RON	03/30/77	09:30	12/01/77	14:13	45	45	26	26	0	9704	26

```
OBJ-LEARN.          LEN  06/28/78 09:28  06/28/78 09:28    45    45     4     4  2  17247    4
TEST48.             LEN  06/29/78 14:32  09/08/78 09:35    20   180    33    16  3   2475   16
MESSAGE-FILE.       LEN  06/28/78 10:02  07/17/78 14:17    20   180     1     4  3   1131    4
OLD-GAT13.          MEL  09/07/78 17:38  01/10/79 16:40    45    45    19    19  4  17454   19
K.                  TRW  08/12/77 17:53  05/25/79 12:29    45    45    48    48  1   3721   48
K/USERS.            TRW  08/17/77 08:21  05/25/79 12:23    90    90    23    40  0   4012   80
OLD-HST16.          MEL  09/07/78 18:04  04/05/79 15:12    45    45    23    23  4   6579   23
LPSINT.             MEL  04/23/79 26:24  04/23/79 24:45    20  1800   131    80  3   5404   80
MOD-TABLE0.         SYST 08/17/77 15:32  08/17/77 15:32    45    45    28    28  0   4329   28
ESMDIAG.            DAN  11/15/77 14:38  05/25/79 12:29    45    45    16    16  1   2937   16
ESMDIAG/CRT.        DAN  05/11/78 14:25  05/25/79 11:15    20  1800   142    80  3  23291   80
ESMDIAG/OBJCRT.     DAN  05/11/78 14:26  05/11/78 14:26    45   180     2     4  0  19698    4
ESMDIAG/RTC.        DAN  11/18/77 15:29  05/25/79 11:16    20   180    31    16  3   2953   16
ESMDIAG/OBJRTC.     DAN  05/11/78 14:11  05/11/78 14:11    45   180     1     4  0  19478    4
ESMDIAG/GAT.        DAN  03/17/78 15:50  04/11/78 15:16    20   180   153    68  3   7270   68
ESMDIAG/OBJ-GAT.    DAN  04/10/78 16:28  04/11/78 09:16    45   180     2     4  0   1356    4
ESMDIAG/DNEX.       DAN  02/16/78 13:28  04/17/78 17:18    20   180    19    12  3   3275   12
ESMDIAG/OBJ-DNEX.   DAN  02/16/78 13:29  05/11/78 15:59    45   180     1     4  0   3287    4
ESMDIAG/OBJ-SDLC-RCV. DAN 04/24/78 18:04  04/27/78 19:40    45   180     2     4  0  17493    4
ESMDIAG/GAT13.      DAN  04/05/79 09:47  04/05/79 17:18    45   180     1     4  0   2394    4
ESMDIAG/SDLC-SND.   DAN  04/24/78 18:02  04/24/78 08:45    20   180   184    84  3   2993   84
ESMDIAG/SDLC-RCV.   DAN  04/24/78 18:03  04/24/78 08:45    20   180   143    64  3   3303   64
ESMDIAG/OBJ-SDLC-SND. DAN 04/24/78 18:04  04/27/78 19:41    45   180     2     4  0   1156    4
OBJ-LPSINT.         MEL  04/23/79 26:25  04/23/79 24:39    45   180     1     4  0   3716    4
CODEF.              MEL  05/25/79 08:10  05/25/79 08:10    45    45    18    18  2   2442   18
DIAG.               DAN  08/22/77 08:33  05/25/79 12:29    45    45    16    16  1  20149   16
DIAG/ACM.           DAN  08/26/77 10:49  10/30/79 13:41    20  1800   142    80  3   3802   80
DIAG/OBJ-ACM.       DAN  05/11/78 14:14  04/23/79 19:52    45   180     2     4  0  19582    4
DIAG/BUF.           MEL  10/14/77 24:21  10/30/79 13:42    20  1800   172    80  3   5496   80
DIAG/OBJ-BUF.       MEL  05/11/78 14:15  04/23/79 19:53    45   180     2     4  0   1077    4
DIAG/RCV.           MEL  02/20/78 37:25  07/17/79 16:31    20  1800   151    80  3   3101   80
DIAG/OBJ-RCV.       MEL  05/25/79 16:08  05/25/79 16:08    45   180     2     4  0   2460    4
DIAG/SEND.          MEL  01/10/78 13:19  07/17/79 16:38    20  1800   111    80  3   6977   80
DIAG/OBJ-SEND.      MEL  05/25/79 16:10  05/25/79 16:10    45   180     1     4  0   2464    4
DIAG/OBJ-CRT.       MEL  03/09/78 18:51  03/18/78 17:17    45   180     2     4  0   1184    4
DIAG/TCCF-RCV.      TRW  04/27/78 17:15  04/27/78 17:16    45   180     1     4  0  17561    4
DIAG/TCCF-SEND.     TRW  04/25/78 15:40  04/27/78 16:48    45   180     1     4  0  17172    4
DIAG/PDP.           DAN  03/23/78 09:23  03/27/78 12:48    20   180    55    28  3   2108   28
DIAG/OBJ-PDP.       DAN  03/23/78 09:33  03/27/78 12:48    45   180     1     4  0   1371    4
EDIT.               MEL  08/18/77 14:18  04/05/79 13:01    45    45   271   271  2  21190  271
MOD-OUTTBL.         MEL  07/21/77 11:18  02/25/78 12:57   256   256     1     6  0   6513    6
PROM.               SYST 02/18/78 60:35  05/25/79 12:29    45    45    48    48  1   3205   48
PROM/LDR16.         MEL  02/21/78 14:55  05/08/78 14:23    20  1800   270   120  3  18131  120
PROM/OBJ-LDR16.     MEL  02/21/78 12:18  05/08/78 14:22    45   180     2     4  0  16884    4
PROM/LDR18.         DAN  02/20/78 60:42  02/21/78 12:32    20  1800   128    80  3   8532   80
PROM/OBJ-LDR18.     DAN  02/20/78 06:58  02/27/78 16:42    45   180     1     4  0   1653    4
PROM/OBJ-LDR12.     DAN  02/21/78 12:22  05/08/78 14:19    45   180     1     4  0  16888    4
PROM/OBJ-LDR13.     DAN  02/21/78 12:24  05/08/78 14:19    45   180     1     4  0  16892    4
PROM/OBJ-LDR14.     DAN  02/21/78 12:25  05/08/78 14:20    45   180     1     4  0  16896    4
PROM/OBJ-LDR15.     DAN  02/21/78 12:26  05/08/78 14:20    45   180     1     4  0  16900    4
PROM/OBJ-LDR17.     DAN  02/21/78 12:31  05/08/78 14:21    45   180     1     4  0  16904    4
PROM/OBJ-LDR19.     DAN  02/21/78 12:32  05/08/78 14:21    45   180     1     4  0  16908    4
LETTERS.            SYST 12/21/77 12:44  05/25/79 12:29    45    45    48    48  1   8718   48
LETTERS/FIN-ESMD1.  DAN  01/02/78 14:21  02/02/78 10:17    20   180    32    16  0   2683   16
MOD-LOADER.         TRW  08/24/77 17:36  02/27/78 11:17    45    45     7     7  2   2013    7
NEW.                MEL  12/01/77 11:10  01/10/79 16:48    20   180     1     4  3   2977    4
GAMES.              RON  12/06/77 08:08  05/25/79 12:29    45    45    32    32  1   5803   32
GAMES/STAR-TREK.    RON  05/25/79 16:44  05/25/79 08:53    20   180    78    36  3   2642   36
GAMES/OTHELLO.      RON  05/25/79 16:46  05/25/79 08:25    20   180    17     8  3   4657    8
GAMES/BLACKJACK.    RON  05/25/79 16:47  05/25/79 16:47    20   180    18     8  3   1425    8
GAMES/NIM.          SYST 06/03/76 11:58  05/25/79 08:29    45    45    27    27  2   3775   27
  216 FILES.  11050 DISK SECTORS.
```

0	***	2
2	SYSTEM/MCP.	
200	SYSTEM/ALGOL.	
325	ALGOL-TABLE.	
500	DIRECTORY.	
635	SYSTEM/BOOTSTRAP.	
640	SYSTEM/STACK.	
800	SYSTEM/DISK-MAP.	
999	SYSTEM/MCP-SEG.	
1000	COLD-START.	
1007	TEST-48HOUR.	
1011	***	2
1013	SYSTEM.	
1061	***	8
1069	OBJ-8776-SEC.	
1072	***	1
1073	CRC.	
1077	DIAG/OBJ-BUF.	
1081	***	2
1083	SYSTEM/LOAD.	
1099	SYSTEM/SEQ-COMPARE.	
1107	SUN-RISE.	
1127	PARSE-OP.	
1131	MESSAGE-FILE.	
1135	CARD-LIST.	
1156	ESMDIAG/OBJ-SDLC-SND.	
1160	SYSTEM/GROUP.	
1166	OBJ-WTDS.	
1170	***	2
1172	SOURCE/CRT.	
1180	OBJ-DSCRT.	
1183	***	1
1184	DIAG/OBJ-CRT.	
1188	SDLC14.	
1208	OBJ-DSCRT16.	
1211	***	6
1217	8721-IAPE-OP.	
1225	ESMD/FTMSG-DAT.	
1237	DSCRT.	
1245	JUNK1.	
1249	***	3
1252	ESMD.	
1268	ESMD/OBJ-MEM.	
1272	SOURCE.	
1320	MEL/MOST-TEST.	
1324	ESMD/GAT13-BDS-ST.	
1356	ESMDIAG/OBJ-GAT.	
1360	***	1
1361	SLAVE.	
1371	DIAG/OBJ-PDP.	
1375	SYSTEM/ASSIGN.	
1395	***	1
1396	WHO.	
1404	***	4
1408	USER/CDIR.	
1412	BDS-TABLE.	
1418	LOOP/SEND-ME.	
1425	GAMES/BLACKJACK.	
1433	***	1
1434	USER/CKTD.	
1438	***	2
1440	SOURCE/WHO.	
1468	MASTER.	
1571	ESMD/LPFT-OBJ.	
1586	TRW/SOURCE.	
1602	***	1
1603	SEQLIST.	
1618	OLD/MST16.	
1641	***	7
1648	MEL/FILE-FIX.	
1653	PROM/OBJ-LDR18.	
1657	***	3
1660	USER.	
1676	***	12
1688	LOOP/BDS-LOADER.	
1695	ESMD/SDLC-BDS-STA.	
1727	SOURCE/NODAL-PATCH.	
1767	***	5
1772	SOURCE/MOST-TEST.	
1788	TRW/SOURCE/TCCF-RCV.	

1812	***	17
1829	ESMD/PDP-BDS-STAC.	
1861	OLD/TCC14.	1
1881	***	
1882	MEL.	
1898	ESMD/BDS-STACK.	
1930	OLD/TCC15.	12
1950	***	
1962	LPFT.	20
1983	***	
2003	OLD/GAT17.	14
2022	***	
2036	DIAGLDR.	1
2047	***	
2048	SYSTEM/DISC-PRINT.	
2056	SYSTEM/MAP-OUT.	16
2062	***	
2078	OLD/ESMLDR.	13
2095	***	
2108	DIAG/PDP.	
2136	OLD/GAT12.	9
2161	***	
2170	OLD/CRT18.	
2193	SOURCE/TABLE.	5
2205	***	
2210	R/USERS.	
2250	ESMLDR.	126
2268	***	
2394	ESMDIAG/GAT13.	
2398	USRLNG.	
2442	CODEF.	
2460	DIAG/OBJ-RCV.	
2464	DIAG/OBJ-SEND.	7
2468	***	
2475	TEST48.	23
2491	***	
2514	ESMD/LPFT-FOR.	36
2606	***	
2642	GAMES/STAR-TREK.	5
2678	***	
2683	LETTERS/FIN-ESMD1.	36
2699	***	
2735	ESMD/GAT13-ESMD-S.	46
2767	***	
2813	USER/INKD.	
2817	TRW/SOURCE/TCCF-SEND.	42
2841	***	
2883	LOGOFF.	50
2887	***	
2937	ESMDIAG.	
2953	ESMDIAG/RTC.	8
2969	***	
2977	NEQ.	12
2981	***	
2993	ESMDIAG/SDLC-SND.	24
3077	***	
3101	DIAG/RCV.	24
3181	***	
3205	PROM.	22
3253	***	
3275	ESMDIAG/DNEX.	
3287	ESMDIAG/OBJ-DNEX.	8
3291	***	
3299	MEL/BDS-LOADER.	
3303	ESMDIAG/SDLC-RCV.	52
3367	***	
3419	SYSTEM/CURRENTSTATU.	
3441	OLD/USRLNG.	232
3484	***	
3716	OBJ-LP5INT.	1
3720	***	
3721	R.	6
3769	***	
3775	GAMES/NIM.	
3802	DIAG/ACM.	30
3882	***	
3912	SYSTEM/DUMP.	
3928	SYSTEM/MT-PRINT.	74
3938	***	
4012	R/USERS.	

4052	***	29
4081	CRC-CALC.	
4093	***	10
4103	MDMPL-TABLE.	
4207	OPERATOR.	
4223	***	64
4287	DEVICE.	
4302	SYSTEM/DIR-LIST.	
4319	***	10
4329	MGB-TABLE0.	
4357	***	60
4417	SYSTEM/MDMPL.	
4488	***	16
4504	LEARN.	
4516	OLD/SDL14.	
4536	OLD/SDL15.	
4556	LENFILE.	
4564	***	62
4626	HST16.	
4649	SOURCE/USRLNG-PATCH.	
4657	GAMES/OTHELLO.	
4665	***	2
4667	RCMAIN.	
4749	JUNK.	
4829	***	55
4884	B776-SEC.	
4964	***	12
4976	USER/INFO.	
5056	SOURCE/NODE-CNTRL.	
5216	***	8
5224	SOURCE/FILE-FIX.	
5244	SOURCE/NODE-CNTRL.	
5404	LPSINT.	
5484	***	12
5496	DIAG/BUF.	
5576	***	100
5676	OLD/ESMD-GAT13-8.	
5708	***	40
5748	OLD/GAT13.	
5774	GAT12.	
5799	USER/TNKD.	
5803	GAMES.	
5835	***	54
5889	SL7-NANO.	
5923	***	54
5977	EE.	
6023	***	71
6094	SOURCE/WTDS.	
6174	***	154
6328	RC.	
6334	***	87
6421	OPERATOR/DEFINES.	
6429	***	84
6513	BDS-OUTTBL.	
6519	TCC14.	
6539	TCC15.	
6559	SDL15.	
6579	OLD-HST16.	
6602	GAT17.	
6621	CRT18.	
6644	SEC19.	
6670	***	133
6803	SOURCE/LOADER-PATCH.	
6807	SOURCE/TEST-48HOUR.	
6823	***	154
6977	DIAG/SEND.	
7057	***	1
7058	SYSTEM/DISK-COMPARE.	
7069	SOURCE/LIST.	
7149	SOURCE/STACK-LIST.	
7229	***	4
7233	SYSTEM/COPY-TOP-BOT.	
7243	***	27
7270	ESMDIAG/GAT.	
7338	ESMD/ESMLDR.	
7498	***	28
7526	USRLN/MSG.	
7606	USER/P0000.	
7766	***	174
7940	USRLN.	

```
7956  ESMD/MEM.
8076  ***                                  70
8146  OLD.
8178  USER/P0000.
8378  MSG.
8412  ***                                  34
8452  USRLN/MSG.
8532  PROM/LDR18.
8612  ***                                 106
8718  LETTERS.
8766  ***                                 102
8868  BDSIM.
8967  USER/P0000.
9127  ESMD/ESMLDR.
9287  LOOP.
9335  ***                                  22
9357  SYSTEM/TRANSLANG.
9455  LOOP/UTILITY.
9496  ***                                  38
9534  SL9-NANO.
9603  ***                                 101
9704  EE-TABLE.
9730  ***                                  39
9769  SYSTEM/DECK-COMPARE.
9777  ***                                   8
9785  SYSTEM/DSIM.
9879  ***                                  23
9902  ESMD/8776-STACK.
9946  ***                                 180
10126 SOURCE/BDS-STACK.
10286 ***                                 368
10654 SOURCE/DUMMY.
10674 GAT13.
10700 SOURCE/STACK-PATCH.
10736 ***                                  41
10777 SOURCE/TAPE-OP.
10937 SOURCE/TAPE-OP.
11097 SOURCE/TAPE-OP.
11257 ***                                 113
11370 CHECK-ATTACH.
11376 ***                                 225
11601 SOURCE/WTDR.
11681 ***                                  38
11719 TRANSLANG-TA.
11810 ***                                 919
12729 TAPE-OP.
12737 ***                                  63
12800 *** GAP                             500
16384 ***
16884 PROM/OBJ-LDR16.
16888 PROM/OBJ-LDR12.
16892 PROM/OBJ-LDR13.
16896 PROM/OBJ-LDR14.
16900 PROM/OBJ-LDR15.
16904 PROM/OBJ-LDR17.
16908 PROM/OBJ-LDR19.
16912 ***                                 260
17172 DIAG/TCCF-SEND.
17176 ***                                  71
17247 OBJ-LEARN.
17251 ***                                  18
17269 SOURCE/NODE-CNTRL.
17429 ***                                  25
17454 OLD-GAT13.
17473 SDL14.
17493 ESMDIAG/OBJ-SDLC-RCV.
17497 ***                                  64
17561 DIAG/TCCF-RCV.
17565 ***                                 186
17751 MEL/CRT-READ.
17754 ***                                 157
17911 SOURCE/CRT-READ.
17919 ***                                  41
17960 SOURCE/BDS-STACK.
18120 ***                                  11
18131 PROM/LDR16.
18251 SOURCE/BDS-STACK.
18411 ***                                1067
19478 ESMDIAG/OBJRTC.
19482 ***                                 100
```

```
19582   DIAG/OBJ-ACM.
19586   ***                              108
19694   OBJ-WTDR.
19698   ESMDIAG/OBJCRT.
19702   ***                              169
19871   SOURCE/BDS-STACK.
20031   ***                               82
20113   BDS-LOADER.
20120   ***                               29
20149   DIAG.
20165   ***                              320
20485   SOURCE/BDS-STACK.
20645   SOURCE/BDS-STACK.
20805   SOURCE/BDS-STACK.
20965   SOURCE/BDS-STACK.
21125   ***                               65
21190   EDIT.
21461   SOURCE/BDS-STACK.
21621   ***                              480
22101   SOURCE/BDS-STACK.
22261   ***                               98
22359   END.
22366   SOURCE/BDS-STACK.
22526   ***                               65
22591   ESMD/LPFT.
22751   ESMD/LPFT.
22911   MSG.
22991   ***                               75
23066   SOURCE/BDS-STACK.
23226   ***                               65
23291   ESMDIAG/CRT.
23371   ***                               74
23445   MEL/TEST-48HOUR.
23449   SOURCE/BDS-STACK.
23609   ***                              618
24227   MGB.
24280   ***                              479
24759   MGB-TABLE.
24830   ***                              818
25648   ESMLORI.
25664   ***                              171
25835   USER/IDIR.
25839   OLD/SEC19.
25865   ***                             3301
29166   TRW.
29182   ***                                2
29184   *** GAP
124 AVAILABLE AREAS.  14548 DISK SECTORS AVAILABLE.
```

4.1 GAT13 Stack Machine

The GAT13 Stack Machine includes an operator to handle the asychronous interface between loop 4 and loop 5. This operator is a modification of the operator used to handle the CRT on loop 4. The differences in the operators are in the areas of character conversion and protocol. In the GAT13 operator the first six characters of a transfer are not converted since they are comprised of the header for the packet and are treated as binary data. In the protocol area, the CRT operator reads and discards a garbage character from the terminal. This function was eliminated in GAT13 stack machine.

AD-A078 392 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66158 SBIE-AD-E100 314 NL

3 OF 4
ADA 078392

```
                        $ MERGE SOURCE/BDS-STACK DISK                                   00000010
001F  00 00 00 00       $ -LIST CLIST                                                   00000100
0027  00 00 00 00
002B  00 00 00 00
002F  00 00 00 00
0DF5  30 31 32 33
0DFD  38 39 23 40
0E05  2B 41 42 43
0E0D  48 49 2E 5B
0E15  5C 4A 4B 4C
0E1D  51 52 24 2A
0E25  20 2F 53 54
0E2D  59 5A 2C 25
0E35  30 0F 3F 0A
0E3D  1D 2D 2B 10
0E45  00 01 02 03
0E4D  08 09 0D 2E
0E55  0B 11 12 13
0E5D  18 19 21 22
0E65  27 28 29 32
0E6D  37 38 39 1B
0EDF  00 DF 0E 1F
                        IOINTERRUPT:                                                    00310400 D
                        BEGIN                           ;*                              00310500 D
                        LABEL INTTEMP, RTCINT, EXCINT, INTEXIT, IOINTERRUPTF,           00310600 D
                           IOINTERRUPTE, INTLOC, LIUINT;                                00310700 D
0EE2  B4                        UMR = M1                ;* SAVE M1                      00310800 D
0EE3  03 ** **                  M1 = INTTEMP            ;* TEMPORARY STORAGE AREA         00310900 D
0EE6  89                        I1 = B0                 ;*                              00311000 D
0EE7  8A                        I1 = B1                 ;*                              00311100 D
0EE8  87                        I1 = WR                 ;*                              00311200 D
                        $OMIT = NOT PIO                                                 00311300 D
                        $POP OMIT                                                       00311600 D
0EE9  8B                        I1 = AD                 ;*                              00311700 D
0EEA  8C                        I1 = J                  ;*                              00311800 D
0EEB  85                        I1 = UMR                ;*                              00311900 D
0EEC  C4                        J = 0                   ;* CLEAR J                       00312000 D
0EED  F3                        B0 = REQ                ;* READ REQUEST LINES             00312100 D
0EEE  CF                        B0D FLIP                ;* FLIP FOR EASY CHECKING         00312200 D
0EEF  E6                        B1 = B0                 ;*                              00312300 D
0EF0  22 ** **                  IF 6S GOTO J + RTCINT   ;*                              00312400 D
0EF3  1D ** **                  IF 7S GOTO J + LIUINT   ;* 7S = CHANNEL 3 = LIU           00312500 D
0EF6  33 ** **                  IF 5R GOTO J + INTEXIT  ;* NOT EXO DEVICE => INVALID DEVICE 00312600 D
0EF9  00 ** **                  GOTO EXOINT             ;*                              00312700 D
0EFC  38 08             LIUINT: AD = #08                ;* LIU CHANNEL #                  00312800 D
0EFE  0C 00                     CIO = #00               ;* SELECT LIU STATUS 0            00312900 D
                        $OMIT = ODD                                                     00313000 D
0F00  40                        SKIP2                   ;*                              00313100 D
                        EXOINT:                                                         00313200 D
0F01  38 02                     AD = #02                ;* EXO CHANNEL #                  00313300 D
                        $POP OMIT                                                       00313400 D
0F03  F2                        B0 = STAT               ;* READ STATUS                   00313500 D
                        INT1:                                                           00313600 D
0F04  03 ** **                  M1 = INTLOC             ;* ADDRESS OF STATUS WORD STORAGE 00313700 D
0F07  8B                        I1 = AD                 ;* STORE CHANNEL ADDRESS          00313800 D
0F08  89                        I1 = B0                 ;*                              00313900 D
0F09  03 54 00                  M1 = INTERPE + OFF      ;* OVERWRITE INTERPE + 1 WITH ... 00314000 D
0F0C  9B                        I1 = 0(1)               ;* GOTO ...                       00314100 D
```

```
0F0D  2F ** **           WR = IOINTERRUPTE          ;*                                  00314200 D
0F10  87                 I1 = WR                    ;* INTERRUPTE                        00314300 D
0F11  03 5F 00           M1 = INTERPF + OFF         ;* OVERWRITE INTERPF + 1 WITH ...     00314400 D
0F14  9B                 I1 = 0(1)                  ;* GOTO ...                          00314500 D
0F15  2F ** **           WR = IOINTERRUPTF          ;*                                  00314600 D
0F18  87                 I1 = WR                    ;* INTERRUPTF                        00314700 D
0F19  2D 04              MASK = #04                 ;* SET MASK TO ALLOW RTC ONLY         00314800 D
                                                                                         00314900 D
                  INTEXIT:
0F1B  03 ** **           M1 = INTTEMP               ;* RESTORE REGISTERS                  00315000 D
0F1E  64                 B0 = I1                    ;*                                  00315100 D
0F1F  57                 B1 = I1                    ;*                                  00315200 D
0F20  5A                 WR = I1                    ;*                                  00315300 D
                  $OMIT = NOT PIO                                                        00315400 D
                  $POP OMIT                                                              00315700 D
0F21  61                 AD = I1                    ;*                                  00315800 D
0F22  68                 J = I1                     ;*                                  00315900 D
0F23  66                 UMR = I1                   ;*                                  00316000 D
0F24  D5                 M1 = UMR                   ;*                                  00316100 D
0F25  B6                 ENABLE RETURN              ;*                                  00316200 D
                                                                                         00316300 D
                  RTCINT:                                                                00316400 D
0F26  38 04              AD = #04                   ;* RTC ADDRESS                       00316500 D
0F28  F2                 B0 = STAT                  ;* RESET RTC FLIP/FLOP               00316600 D
0F29  03 ** **           M1 = TIMEWORD              ;*                                  00316700 D
0F2C  5A                 WR = I1                    ;*                                  00316800 D
0F2D  04 01 00           WR = WR + 1                ;*                                  00316900 D
0F30  AC AC              M1 = M1 - 2                ;*                                  00317000 D
0F32  87                 I1 = WR                    ;*                                  00317100 D
0F33  B0                 Z = WR - 1                 ;*                                  00317200 D
0F34  25 1B 0F           IF KS GOTO INTEXIT         ;*                                  00317300 D
0F37  5A                 WR = I1                    ;*                                  00317400 D
0F38  04 01 00           WR = WR + 1                ;*                                  00317500 D
0F3B  AC AC              M1 = M1 - 2                ;*                                  00317600 D
0F3D  87                 I1 = WR                    ;*                                  00317700 D
0F3E  00 1B 0F           GOTO INTEXIT               ;*                                  00317800 D
                  $OMIT = NOT PIO                                                        00317900 D
                  $POP OMIT + OMIT = NOT SDLC                                            00318700 D
                  $POP OMIT                                                              00322100 D
0F41  00 00              TIMEWORD: PLANTD1 0;       ;*                                  00322200 D
0F43  00 00                      PLANTD1 0          ;*                                  00322300 D
0F45  00                         PLANT1 0           ;*                                  00322400 D
0F46  00                 INTLOC: PLANT1  0          ;* CHANNEL NUMBER                    00322500 D
0F47  00                         PLANT1 0           ;* STATUS WORD                       00322600 D
0F48  00 00 00 00        INTTEMP:ZERO8 ;ZERO4; PLANT1 0  ;* TEMPORARY STORAGE (13  BYTES)  00322700 D
0F54  00 00 00 00
                  IOINTERRUPTF:                                                          00322800 D
0F55  82                 I2 = WR                    ;* STORE SECOND HALF OF TOS          00322900 D
                  IOINTERRUPTE:                                                          00323000 D
0F56  34 00 00           L = M2 + 0                 ;* SAVE STACK POINTER                 00323100 D
0F59  AC                 M1 = M1 - 1                ;* A+JUST PC                          00323200 D
0F5A  07 54 00           M2 = INTERPE + OFF         ;* OVERWRITE INTERPE + 1 WITH         00323300 D
0F5D  3A 1D              B0 = #1D                   ;*                                  00323400 D
0F5F  94                 I2 = B0                    ;* IF 7S GOTO J + ...                 00323500 D
0F60  2F BD 00           WR = OPXE                  ;*                                  00323600 D
0F63  82                 I2 = WR                    ;* OPXE                              00323700 D
0F64  07 5F 00           M2 = INTERPF + OFF         ;* OVERWRITE INTERPF + 1 WITH         00323800 D
0F67  3A 1D              B0 = #1D                   ;*                                  00323900 D
0F69  94                 I2 = B0                    ;* IF 7S GOTO J + ...                 00324000 D
0F6A  2F CB 00           WR = OPXF                  ;*                                  00324100 D
0F6D  82                 I2 = WR                    ;* OPXF                              00324200 D
```

```
0F6E  07 ** **          M2 = STACKBASE + 140      ;* I/O INTERRUPT PCW                  00324300 D
0F71  56                WR = I2                   ;*                                   00324400 D
0F72  69                B32 = I2                  ;*                                   00324500 D
0F73  12 00 00          M2 = L + 0                ;*                                   00324600 D
0F76  82                I2 = WR                   ;* PUSH PCW ONTO STACK               00324700 D
0F77  83                I2 = B32                  ;*                                   00324800 D
0F78  2E 50 07          CALL MARKSTACK            ;*                                   00324900 D
0F7B  34 FE FF          L = M2 - 2                ;* SAVE STACK POINTER                00325000 D
0F7E  07 46 0F          M2 = INTLOC               ;* ADDRESS OF INTERRUPT WORD         00325100 D
0F81  69                B32 = I2                  ;* INTERRUPT WORD                    00325200 D
0F82  12 02 00          M2 = L + 2                ;*                                   00325300 D
0F85  84                I2 = 0(2)                 ;* CLEAR UH TOS                      00325400 D
0F86  34 00 00          L = M2 + 0                ;*                                   00325500 D
0F89  0C 96 07          GOTO NTR                  ;*                                   00325600 D
                  END                             ;*                                   00325700 D
                  BEGIN                           ;*                                   00325800 D
                  LABEL LIO1, LIO2, LIO3, WRITE8, CRCOK, LIOEXIT, LIO4, LIO5,          00325900 D
                     LIO6, LIO7, LIO8, LIO9, LIO10, WRITE8A;                           00326000 D
                  LABEL STAT2, STAT1, WTKN2, WTKN3, RAM1, RESETLOOP,                   00326100 D
                     BUFRESET, BUFRESET1, RAM2;                                        00326200 D
                  LABEL ADIN0=#08, ADIN1=#88, DIN0=#20,  DIN1=#A0, CRCMSK0=#01,        00326300 D
                     CRCMSK1=#02, MODSTAT=#10, OB0FULL=#01, OB1FULL=#02,RDSTAT1=#80,   00326400 D
                     RDSTAT0=#00, DOUT0=#60, DOUT1=#E0, ADOUT0=#48, ADOUT1=#C8,        00326500 D
                     OB1FMSK=#08, ACRAD=#02, ACRAM=#01, NULLRAM=#07, READ=#04, READND=#06,00326600 D
                     WTOKEN=#00, WRTCMD=#10, OBSFULL=#03;                              00326700 D
                  *                                                                    00326800 D
                  *   LIU INPUT-OUTPUT HANDLER     LIO (BUFFER, MEMORY ADDRESS, SIZE)  00326900 D
                  *                                                                    00327000 D
0F8C  12 00 00    LIO:  M2 = L + 0                ;*                                   00327100 D
0F8F  83                I2 = B32                  ;* PUSH                             00327200 D
0F90  12 F9 FF          M2 = L - 7                ;* POINT TO BUFFER NUMBER            00327300 D
0F93  53                B0 = I2                   ;* B0 = BUFFER NUMBER                00327400 D
0F94  56                WR = I2                   ;* JUNK                              00327500 D
0F95  56                WR = I2                   ;* MEMORY ADDRESS                    00327600 D
0F96  AD                WR FLIP                   ;*                                   00327700 D
0F97  A1                WR = WR ! 2               ;* MAKE BYTE ADDR                    00327800 D
0F98  A1                WR = WR ! 2               ;*                                   00327900 D
0F99  E0                XY = WR(2)                ;* SAVE ADDR                         00328000 D
0F9A  E0                XY = WR(2)                ;*                                   00328100 D
0F9B  54                B1 = I2                   ;* SIGN OF SIZE (R/W)                00328200 D
0F9C  B8                B1 = B1 V B0              ;* INSERT BUF # & CONTINUE BIT        00328300 D
0F9D  51                WRL = I2                  ;* JUNK                              00328400 D
0F9E  56                WR = I2                   ;* SIZE                              00328500 D
0F9F  AD                WR FLIP                   ;*                                   00328600 D
0FA0  E0                XY = WR(2)                ;*                                   00328700 D
0FA1  E0                XY = WR(2)                ;*                                   00328800 D
0FA2  BB                K = M1                    ;* SAVE PC                          00328900 D
0FA3  38 08             AD = #08                  ;* CHANNEL ADDRESS                   00329000 D
0FA5  E1                WR = XY                   ;* RECALL MEMORY ADDRESS             00329100 D
0FA6  E1                WR = XY                   ;*                                   00329200 D
0FA7  A4                M1 = WR                   ;*                                   00329300 D
0FA8  40 FF FF          Z = WR + #FFFF            ;* TEST FOR ADDRESS = 0              00329400 D
0FAB  26 ** **          IF KR GOTO LIO5           ;*                                   00329500 D
0FAE  0E 80             B1 = B1 V #80             ;* SET KEEP DATA FLAG                00329600 D
                  LIO5:                                                                00329700 D
0FB0  E1                WR = XY                   ;* RECALL SIZE                       00329800 D
0FB1  E1                WR = XY                   ;*                                   00329900 D
0FB2  22 ** **          IF 6S GOTO J + WRITE8     ;* 6S=>WRITE                         00330000 D
0FB5  3A 08             B0 = ADIN0                ;* INPUT BUFFER 0 ADDRESS COMMAND    00330100 D
0FB7  20 ** **          IF 0R GOTO J + LIO1       ;*                                   00330200 D
```

```
0FFA  3A 88              B0 = ADIN1                ;* INPUT BUFFER 1 ADDRESS COMMAND     00330300 D
                   LIO1:                                                                  00330400 D
0FBC  EB                 CIO = B0                  ;*                                   00330500 D
0FBD  F0                 B0 = IO                   ;* READ BUFFER ADDRESS                00330600 D
0FBE  24 ** **           IF 1R GOTO J + LIO10      ;* IF CONTINUATION, READ             00330700 D
0FC1  35 01              B0 = B0 + 1               ;* ....1 MORE CHAR OF DATA           00330800 D
                   LIO10:                                                                 00330900 D
0FC3  C5                 B1FL FLIP                 ;* SAVE BYTE FLAG                     00331000 D
0FC4  E6                 B1 = B0                   ;*                                   00331100 D
0FC5  01 FF              Z = WRL + #FF             ;* TEST FOR SIZE = 0                  00331200 D
0FC7  26 ** **           IF KR GOTO LIO4           ;*                                   00331300 D
0FCA  10 FF              B0 = B1 \ #FF             ;* B0 = ~ PTR - 1                     00331400 D
0FCC  35 01              B0 = B0 + 1               ;* B0 = -PTR                         00331500 D
0FCE  D0                 B0 = B0 + WRL             ;* B0 = WRL - PTR                    00331600 D
0FCF  26 ** **           IF KR GOTO LIO8           ;* KR => USE SIZE                    00331700 D
                   LIO4:                                                                  00331800 D
0FD2  D3                 WRL = B1                  ;*                                   00331900 D
0FD3  C5                 B1FL FLIP                 ;*                                   00332000 D
0FD4  0E 40              B1 = B1 V #40             ;* SET EMPTY FLAG                     00332100 D
0FD6  02                 SKIP1                     ;*                                   00332200 D
                   LIO8:                                                                  00332300 D
0FD7  C5                 B1FL FLIP                 ;*                                   00332400 D
0FD8  3A 20              B0 = DIN0                 ;* READ BUFFER 0 COMMAND              00332500 D
0FDA  20 ** **           IF 0R GOTO J + LIO3       ;*                                   00332600 D
0FDD  3A A0              B0 = DIN1                 ;* RAAD BUFFER 1 COMMAND              00332700 D
                   LIO3:                                                                  00332800 D
0FDF  EB                 CIC = B0                  ;*                                   00332900 D
0FE0  24 ** **           IF 1R GOTO J + LIO9       ;* IF CONTINUATION READ              00333000 D
0FE3  02                 SKIP1                     ;* DON T DO DUMMY READ               00333100 D
                   LIO9:                                                                  00333200 D
0FE4  F0                 B0 = IO                   ;* DUMMY READ                         00333300 D
0FE5  1A ** **           IF 7R GOTO J + LIO6       ;* 7R => IGNORE DATA                 00333400 D
0FE8  C6                 TX = WRL                  ;* SIZE                               00333500 D
0FE9  90                 I1 = IO                   ;*                                   00333600 D
                   LIO7:                                                                  00333700 D
0FEA  0C 80              CIO = RDSTAT1             ;*                                   00333800 D
0FEC  F2                 B0 = STAT                 ;* READ STATUS                        00333900 D
0FED  20 ** **           IF 0R GOTO J + LIO2       ;*                                   00334000 D
0FF0  15 02              B0 = B0 & CRCMSK1         ;* ISOLATE CRC1OK                     00334100 D
0FF2  40                 SKIP2                     ;*                                   00334200 D
                   LIO2:                                                                  00334300 D
0FF3  15 01              B0 = B0 & CRCMSK0         ;*                                   00334400 D
0FF5  AD                 WR FLIP                   ;*                                   00334500 D
0FF6  31 00              Z = B0 = #00              ;*                                   00334600 D
0FF8  26 ** **           IF KR GOTO CRCOK          ;*                                   00334700 D
0FFB  3B 55              WRL = #00                 ;* SET RESULT DESCRIPTOR FALSE       00334800 D
0FFD  40                 SKIP2                     ;*                                   00334900 D
                   CRCOK:                                                                 00335000 D
0FFE  3B 54              WRL = #01                 ;* SET RESULT DESCRIPTOR TRUE        00335100 D
1000  1B ** **           IF 6R GOTO J + LIOEXIT    ;* 6R => BUFFER NOT EMPTY            00335200 D
1003  DB                 B0 = WRL                  ;*                                   00335300 D
1004  2B 02              B0 = B0 V 2               ;*                                   00335400 D
1006  A5                 WRL = B0                  ;*                                   00335500 D
                   LIOEXIT:                                                               00335600 D
1007  AD                 WR FLIP                   ;*                                   00335700 D
1008  12 F6 FF           M2 = L - 10               ;*                                   00335800 D
100B  84                 I2 = 0(2)                 ;*                                   00335900 D
100C  C2                 M1 = K                    ;* RESET PC                          00336000 D
100D  00 5E 00           GOTO INTERPF              ;*                                   00336100 D
                   LIO6:                                                                  00336200 D
```

```
1010  C6                TX = WRL                ;*                                00336300 D
1011  F0                B0 = IO                 ;*                                00336400 D
1012  00 EA 0F          GOTO LI07               ;*                                00336500 D
                  *                                                               00336600 D
                  INTENABLE:                                                      00336700 D
1015  B6                ENABLE RETURN           ;*                                00          *
                                                  00336900 D
                  *                                                               00337000 D
                  WRITE8:                                                         00337100 D
1016  0C 60             CIO = DOUT0             ;*                                00337200 D
1018  C6                TX = WRL                ;* SIZE                           00337300 D
1019  5F                IO = I1                 ;*                                00337400 D
101A  A0                WRU = 0                 ;*                                00337500 D
101B  24 ** **          IF 1R GOTO J + WRITE8A  ;*                                00337600 D
101E  00 FE 0F          GOTO CRCOK              ;*                                00337700 D
                  WRITE8A:                                                        00337800 D
1021  0C E0             CIO = DOUT1             ;*                                00337900 D
1023  17 00             IO = 0                  ;* CONTROL CHAR                   00338000 D
1025  17 FF             IO = #FF                ;* WRITE TOKEN ADDRESS            00338100 D
1027  0C 10             CIO = MODSTAT           ;*                                00338200 D
1029  17 03             IO = OBSFULL            ;*                                00338300 D
102B  00 FE 0F          GOTO CRCOK              ;* EXIT                           00338400 D
                  *                                                               00338500 D
                  *     WRITE ADDRESS COMPARISON RAM    RAM (TYPE/ADDR)           00338600 D
                  *                                                               00338700 D
                  RAM:                                                            00338800 D
102E  38 08             AD = #08                ;*                                00338900 D
1030  ED                B1 = B2                 ;* TYPE                           00339000 D
1031  39 FF             B1 = B1 \ #FF           ;* COMPLEMENT TYPE               00339100 D
1033  F5                B32 FLIP                ;*                                00339200 D
1034  EC                B0 = B2                 ;* ADDR RAM ADDRESS               00339300 D
1035  0C 02             CIO = ACRAD             ;* LOAD ADDRESS COMMAND           00339400 D
1037  EA                IO = B0                 ;* ADDR1                          00339500 D
1038  3A 06             B0 = READND             ;* RAAD NON-DESTRUCTIVE VALUE     00339600 D
103A  20 ** **          IF 0R GOTO J + RAM1     ;*                                00339700 D
103D  3A 04             B0 = READ               ;* READ VALUE                     00339800 D
103F  24 ** **          IF 1R GOTO J + RAM1     ;*                                00339900 D
1042  3A 00             B0 =  WTOKEN            ;* WRITE TOKEN DETECT VALUE       00340000 D
1044  2A ** **          IF 2R GOTO J + RAM1     ;*                                00340100 D
1047  3A 07             B0 = NULLRAM            ;* N LL VALUE                     00340200 D
1049  1F ** **          IF 3S GOTO J + RAM1     ;*                                00340300 D
104C  00 ** **          GOTO RAM2               ;*                                00340400 D
                  RAM1:                                                           00340500 D
104F  0C 01             CIO =  ACRAM             ;*                               00340600 D
1051  EA                IO = B0                 ;*                                00340700 D
1052  12 00 00          M2 = L + 0              ;*                                00340800 D
1055  00 5E 00          GOTO INTERPF            ;*                                00340900 D
                  RAM2:                                                           00341000 D
1058  F0                B0 = IO                 ;* READ RAM                       00341100 D
1059  A0                WRU = 0                 ;*                                00341200 D
105A  A5                WRL = B0                ;*                                00341300 D
105B  AD                WR FLIP                 ;*                                00341400 D
105C  12 00 00          M2 = L + 0              ;*                                00341500 D
105F  00 5E 00          GOTO INTERPF            ;*                                00341600 D
                  *                                                               00341700 D
                  *     READ STATUS       STAT (CHAN)                             00341800 D
                  *                                                               00341900 D
                  STATUS:                                                         00342000 D
1062  F5                B32 FLIP                ;*                                00342100 D
1063  EC                B0 = B2                 ;* CHAN                           00342200 D
```

```
1064  E3                        AD = B0                    ;*                                  00342300 D
1065  31 08                     Z = B0 = #08               ;*                                  00342400 D
1067  26 ** **                  IF KR GOTO STAT2           ;* KR => CHAN = 2                    00342500 D
106A  0C 80                     CIO = RDSTAT1              ;* READ STATUS COMMAND FOR LIU       00342600 D
106C  F2                        B0 = STAT                  ;* BYTE1                           00342700 D
106D  A5                        WRL = B0                   ;*                                  00342800 D
106E  0C 00                     CIO = RDSTAT0              ;*                                  00342900 D
1070  F2                        B0 = STAT                  ;* BYTE0                           00343000 D
1071  AD                        WR FLIP                    ;*                                  00343100 D
1072  A5                        WRL = B0                   ;*                                  00343200 D
1073  12 00 00                  M2 = L + 0                 ;*                                  00343300 D
1076  00 5E 00                  GOTO INTERPF               ;*                                  00343400 D
                    STAT2:                                                                    00343500 D
1079  F2                        B0 = STAT                  ;*                                  00343600 D
                    STAT1:                                                                    00343700 D
107A  A5                        WRL = B0                   ;* PUT STATUS WORD ON TOS         00343800 D
107B  A0                        WRU = 0                    ;*                                  00343900 D
107C  AD                        WR FLIP                    ;*                                  00344000 D
107D  12 00 00                  M2 = L + 0                 ;*                                  00344100 D
1080  00 5E 00                  GOTO INTERPF               ;*                                  00344200 D
                    *                                                                         00344300 D
                    *       SET BDS LINE SWITCH BITS   SWTCH (CODE)                           00344400 D
                    *                                                                         00344500 D
                    SWTCH:                                                                    00344600 D
1083  F5                        B32 FLIP                   ;*                                  00344700 D
1084  EC                        B0 = B2                    ;* CODE                           00344800 D
1085  38 08                     AD = #08                   ;* LIU CHANNEL                    00344900 D
1087  0C 10                     CIO = MODSTAT              ;*                                  00345000 D
1089  EA                        IO = B0                    ;*                                  00345100 D
108A  12 00 00                  M2 = L + 0                 ;*                                  00345200 D
108D  00 5E 00                  GOTO INTERPF               ;*                                  00345300 D
                    $ OMIT = NOT ESMD                                                         00345400 D
                    *                                                                         00345500 D
                    *    READ REAL TIME CLOCK    TIME (0)                                     00345600 D
                    *                                                                         00345700 D
                    TIME:                                                                     00345800 D
1090  38 04                     AD = #04                   ;*                                  00345900 D
1092  F0                        B0 = IO                    ;*                                  00346000 D
1093  07 41 0F                  M2 = TIMEWORD              ;*                                  00346100 D
1096  A5                        WRL = B0                   ;*                                  00346200 D
1097  AD                        WR FLIP                    ;*                                  00346300 D
1098  51                        WRL = I2                   ;*                                  00346400 D
1099  69                        B32 = I2                   ;*                                  00346500 D
109A  12 FE FF                  M2 = L - 2                 ;*                                  00346600 D
109D  F5                        B32 FLIP                   ;*                                  00346700 D
109E  83                        I2 = B32                   ;*                                  00346800 D
109F  00 5E 00                  GOTO INTERPF               ;*                                  00346900 D
                    $ POP OMIT                                                                00347000 D
                    *                                                                         00347100 D
                    *      ISSUE WRITE TOKEN     WTOKEN (WTA)                                 00347200 D
                    WTKN:                                                                     00347300 D
10A2  38 08                     AD = #08                   ;* LIU ADDRESS                    00347400 D
10A4  0C 80                     CIO = RDSTAT1              ;* READ STATUS                     00347500 D
10A6  B5                        B1 = STAT                  ;*                                  00347600 D
10A7  1F ** **                  IF 3S GOTO J + WTKN2       ;* IF OB1F SEND WRITE COMMAND      00347700 D
10AA  0C E0                     CIO = DOUT1                ;*                                  00347800 D
10AC  17 00                     IO = 0                     ;* CONTROL CHARACTER               00347900 D
10AE  17 FF                     IO = #FF                   ;* WTA                             00348000 D
10B0  0C 10                     CIO = MODSTAT              ;*                                  00348100 D
10B2  17 02                     IO = OB1FULL               ;*                                  00348200 D
```

```
                        WTKN2:                                                                  00348300 D
10B4  0C 10                     CIO = MODSTAT           ;*                                      00348400 D
10B6  17 10                     IO = WRTCMD             ;* ISSUE WRITE COMMAND                  00348500 D
                        WTKN3:                                                                  00348600 D
10B8  0C 80                     CIO = RDSTAT1           ;*                                      00348700 D
10BA  F2                        B0 = STAT               ;* READ STATUS                          00348800 D
10BB  15 08                     B0 = B0 & OB1FMSK       ;* ISOLATE OB1F                        00348900 D
10BD  31 00                     Z = B0 = #00            ;*                                      00349000 D
10BF  26 B8 10                  IF KR GOTO WTKN3        ;* IF OB1F = 1, LOOP                    00349100 D
10C2  12 00 00                  M2 = L + 0              ;*                                      00349200 D
10C5  2F 00 01                  WR = #0100              ;*                                      00349300 D
10C8  00 5E 00                  GOTO INTERPF            ;*                                      00349400 D
                        *                                                                       00349500 D
                        *                                                                       00349600 D
                        *    LIU INITIALIZATION ROUTINE                                         00349700 D
                        *                                                                       00349800 D
                        LIUINIT:                                                                00349900 D
10CB  38 08                     AD = #08                ;*                                      00350000 D
10CD  0C 10                     CIO = MODSTAT           ;*                                      00350100 D
                        $ OMIT = NOT ESMD                                                       00350200 D
10CF  17 0C                     IO = #0C                ;* SET LINE SWITCH BITS                 00350300 D
                        $ POP OMIT                                                              00350400 D
10D1  17 60                     IO = #60                ;* RESET STATUS FLIP-FLOPS              00350500 D
10D3  37 AB                     B1 = 0                  ;*                                      00350600 D
                        RESETLOOP:                                                              00350700 D
10D5  2E ** **                  CALL BUFRESET           ;*                                      00350800 D
10D8  42 40                     B1 = B1 + #40           ;* INCREMENT BUFFER SELECTOR            00350900 D
10DA  26 D5 10                  IF KR GOTO RESETLOOP    ;*                                      00351000 D
10DD  2D 00                     MASK = #00              ;*                                      00351100 D
10DF  0C 00                     CIO = RDSTAT0           ;* CLEAR STATUS BYTE 0                  00351200 D
                        $OMIT = NOT PIO                                                         00351300 D
                        $POP OMIT                                                               00352000 D
-                      $POP OMIT                                                                00353100 D
10E1  B6                        ENABLE RETURN           ;*                                      00353200 D
                        BUFRESET:                                                               00353300 D
10E2  DC                        B0 = B1                 ;* BUFFER SELECT                        00353400 D
10E3  2B 08                     B0 = B0 V #08           ;*                                      00353500 D
10E5  EB                        CIO = B0                ;* READ BUFFER ADDRESS COMMAND          00353600 D
10E6  F0                        B0 = IO                 ;* RAAD BUFFER ADDRESS                  00353700 D
10E7  A5                        WRI = B0                ;*                                      00353800 D
10E8  DC                        B0 = B1                 ;*                                      00353900 D
10E9  2B 20                     B0 = B0 V #20           ;*                                      00354000 D
10EB  EB                        CIO = B0                ;* READ BUFFER COMMAND                   00354100 D
10EC  C6                        TX = WRL                ;*                                      00354200 D
10ED  F0                        B0 = IO                 ;* RESET BUFFER POINTER TO ZERO          00354300 D
10EE  22 ** **                  IF 6S GOTO J + BUFRESET1 ;*                                     00354400 D
10F1  F0                        B0 = IO                 ;* SET INPUT BUFFER POINTER TO FF        00354500 D
10F2  A6                BUFRESET1: RETURN               ;*                                      00354600 D
                        *                                                                       00354700 D
                        END     ;*                                                              00354800 D
                        $ OMIT = NOT ESMD                                                       00354900 D
                        *                                                                       00355000 D
                        *       GATEWAY INPUT           OUTPUT HANDLER                          00355100 D
                        *                                                                       00355200 D
                        BEGIN                           ;*                                      00355300 D
                        LABEL GIOEXIT, GIOEXITF, GIOREAD, GIOWRITE, GIOWRITE1, READDATA, READ3, 00355400 D
                           READRELOC, WRITEDATA, WRITE1, WRITE2, WRITE3, GIO1, GIOFLAG,         00355500 D
                           WRITELOOP, TRANSMIT;                                                 00355600 D
                        *                                                                       00355700 D
```

```
                        GIO:                            ;* GIO (BUFFER, SIZE)                00355800 D
10F3  07 ** **                  M2 = GIOFLAG            ;*                                  00355900 D
10F6  53                        B0 = I2                 ;*                                  00356000 D
10F7  26 ** **                  IF KR GOTO GIO1         ;* CHECK IF INTERFACE INITIALIZED    00356100 D
10FA  F5                        B32 FLIP                ;*                                  00356200 D
10FB  CA                        B32 = B32 ! 2           ;* MAKE BYTE SIZE                     00356300 D
10FC  CA                        B32 = B32 ! 2           ;* MAKE BYTE SIZE                     00356400 D
10FD  12 FC FF                  M2 = L - 4              ;* POINTER BUFFER                     00356500 D
1100  56                        WR = I2                 ;*                                  00356600 D
1101  AD                        WR FLIP                 ;*                                  00356700 D
1102  A1                        WR = WR ! 2             ;* MAKE BYTE ADDRESS                  00356800 D
1103  A1                        WR = WR ! 2             ;* MAKE BYTE ADDRESS                  00356900 D
1104  38 02                     AD = #02                ;* DEVICE ADDRESS                     00357000 D
1106  0C 08                     CIC = #08               ;* FRZ STATUS REG                     00357100 D
1108  12 FE FF                  M2 = L - 2              ;*                                  00357200 D
110B  54                        B1 = I2                 ;* READ SIGN                          00357300 D
110C  22 ** **                  IF 6S GOTO J + GICWRITE ;* NEG SIGN => WRITE                 00357400 D
110F  B5                        B1 = STAT               ;* READ OB STATUS                     00357500 D
1110  39 FF                     B1 = B1 \ #FF           ;*                                  00357600 D
1112  F2                        B0 = STAT               ;* RESTORE STATUS REGISTER            00357700 D
1113  0C 00                     CIO = #00               ;* STATUS RESUME                      00357800 D
1115  20 ** **                  IF 0R GOTO J + GIOREAD  ;* 0R => OB FULL                     00357900 D
                        GIOEXITF:                                                           00358000 D
1118  2F 00 00                  WR = #0000              ;* SET FLAG FOR FALSE                 00358100 D
                        GIOEXIT:                                                            00358200 D
111B  12 FA FF                  M2 = L - 6              ;*                                  00358300 D
111E  84                        I2 = 0(2)               ;*                                  00358400 D
111F  00 5E 00                  GOTO INTERPF            ;*                                  00358500 D
1122  0C 10             GIOREAD: CIO = #10              ;*                                  00358600 D
1124  0C 00                     CIO = #00               ;*                                  00358700 D
1126  0C 01                     CIO = #01               ;* READY FOR TRANSFER                00358800 D
1128  BB                        K = M1                  ;* SAVE PC                           00358900 D
1129  A4                        M1 = WR                 ;* BUFFER MEMORY                      00359000 D
112A  05 FF                     TX = 255                ;*                                  00359100 D
112C  90                        I1 = IO                 ;* READ DATA                          00359200 D
112D  0A 07 FF                  M1 = M1 - 249           ;*                                  00359300 D
1130  3C 06 00                  B32 = 6                 ;* SET COUNT                          00359400 D
                        READDATA:                                                           00359500 D
1133  64                        B0 = I1                 ;* READ CHARACTER                    00359600 D
1134  25 ** **                  IF KS GOTO READ3        ;* IF LEOP ,LEAVE ALONE               00359700 D
1137  15 7F                     B0 = B0 & #7F           ;* STRIP OFF PARITY BIT               00359800 D
1139  02 E0                     Z = B0 - #20            ;*                                  00359900 D
113B  19 ** **                  IF KR GOTO J + READRELOC ;* ADD #40                         00360000 D
113E  02 A0                     Z = B0 - #60            ;*                                  00360100 D
1140  18 ** **                  IF KS GOTO J + READ3    ;*                                  00360200 D
1143  35 E0                     B0 = B0 - #20           ;*                                  00360300 D
1145  A5                        WRL = B0                ;*                                  00360400 D
1146  A0                        WRU = 0                 ;*                                  00360500 D
1147  06 35 0E                  M2 = WRR + ASCINT       ;*                                  00360600 D
114A  53                        B0 = I2                 ;*                                  00360700 D
114B  40                        SKIP2                   ;*                                  00360800 D
                        READRELOC:                                                          00360900 D
114C  35 40                     B0 = B0 + #40           ;*                                  00361000 D
                        READ3:                                                              00361100 D
114E  11 01 00                  B32 = B32 + 1           ;*                                  00361200 D
1151  AC                        M1 = M1 - 1             ;*                                  00361300 D
1152  89                        I1 = B0                 ;*                                  00361400 D
1153  19 33 11                  IF KR GOTO J + READDATA ;*                                  00361500 D
1156  0C 00                     CIO = #00               ;*                                  00361600 D
1158  0C 10                     CIO = #10               ;* RESET BAR & BC                    00361700 D
```

```
115A  0C 00              CIC = #00                  ;*                                00361800 D
115C  C2                 M1 = K                     ;* RESTORE PC                     00361900 D
115D  2F 00 01           WR = #0100                 ;* SET FLAG FOR TRUE               00362000 D
1160  C9                 WR = WR + B32              ;* ADD IN SIZE OF TRANSFER          00362100 D
1161  00 1B 11           GOTO GIOEXIT               ;*                                00362200 D
                   GIOWRITE:                                                           00362300 D
1164  F2                 B0 = STAT                  ;* DISCARD OB STATUS               00362400 D
1165  D7                 NOOP                       ;*                                00362500 D
1166  B5                 B1 = STAT                  ;* READ IB STATUS                  00362600 D
1167  39 FF              B1 = B1 \ #FF              ;*                                00362700 D
1169  0C 00              CIO = #00                  ;*                                00362800 D
116B  20 ** **           IF 0R GOTO J + GIOWRITE1   ;*                                00362900 D
116E  00 18 11           GOTO GIOEXITF              ;*                                00363000 D
                   GIOWRITE1:                                                          00363100 D
1171  BB                 K = M1                     ;* SAVE PC                         00363200 D
1172  A4                 M1 = WR                    ;*                                00363300 D
1173  3B AA              WRL = #FF                  ;* MASK FOR COMPLEMENTING CHAR     00363400 D
1175  37 A5              B1 = #FA                   ;* LOOP COUNTER                     00363500 D
                   WRITELOOP:                                                          00363600 D
1177  64                 B0 = I1                    ;* READ CHAR                       00363700 D
1178  E5                 B0 = B0 \ WRL              ;* COMPLEMENT CHAR                 00363800 D
1179  AC                 M1 = M1 - 1                ;*                                00363900 D
117A  89                 I1 = B0                    ;* WRITE CHAR TO OB                00364000 D
117B  42 01              B1 = B1 + 1                ;*                                00364100 D
117D  26 77 11           IF KR GOTO WRITELOOP       ;*                                00364200 D
1180  11 FA FF           B32 = B32 - 6              ;* ADJUST COUNT                     00364300 D
                   WRITEDATA:                                                          00364400 D
1183  B1                 B32 = B32 - 1              ;*                                00364500 D
1184  19 ** **           IF KR GOTO J + TRANSMIT    ;*                                00364600 D
                   *                                                                   00364700 D
                   *                                                                   00364800 D
                   WRITE1:                                                             00364900 D
1187  64                 B0 = I1                    ;*                                00365000 D
1188  02 A0              Z = B0 - #60               ;*                                00365100 D
118A  18 ** **           IF KS GOTO J + WRITE3      ;* NO CHANGE                       00365200 D
118D  02 C0              Z = B0 - #40               ;*                                00365300 D
118F  18 ** **           IF KS GOTO J + WRITE2      ;* RELOC                           00365400 D
1192  A5                 WRL = B0                   ;*                                00365500 D
1193  A0                 WRU = 0                    ;*                                00365600 D
1194  06 F5 0D           M2 = WRR + INTASC          ;*                                00365700 D
1197  53                 B0 = I2                    ;*                                00365800 D
1198  40                 SKIP2                      ;*                                00365900 D
                   WRITE2:                                                             00366000 D
1199  35 C0              B0 = B0 - #40              ;*                                00366100 D
                   WRITE3:                                                             00366200 D
119B  3B AA              WRL = #FF                  ;*                                00366300 D
119D  E5                 B0 = B0 \ WRL              ;* COMPLEMENT CHARACTER            00366400 D
119E  AC                 M1 = M1 - 1                ;*                                00366500 D
119F  89                 I1 = B0                    ;*                                00366600 D
11A0  00 83 11           GOTO WRITEDATA             ;*                                00366700 D
                   TRANSMIT:                                                           00366800 D
11A3  12 FC FF           M2 = L - 4                 ;* POINTER BUFFER                  00366900 D
11A6  56                 WR = I2                    ;*                                00367000 D
11A7  AD                 WR FLIP                    ;*                                00367100 D
11A8  A1                 WR = WR ! 2                ;* MAKE BYTE ADDRESS               00367200 D
11A9  A1                 WR = WR ! 2                ;* MAKE BYTE ADDRESS               00367300 D
11AA  A4                 M1 = WR                    ;*                                00367400 D
11AB  0C 02              CIO = #02                  ;* PUT GIB INTO LOA MODE           00367500 D
11AD  05 FF              TX = 255                   ;*                                00367600 D
11AF  5F                 IO = I1                    ;*                                00367700 D
```

```
11B0  0C 00             CIO = #00                ;*                                  00367800 D
11B2  0C 04             CIO = #04                ;* START TRANSMISSION               00367900 D
11B4  0C 00             CIO = #00                ;*                                  00368000 D
11B6  C2                M1 = K                   ;* RESTORE PC                        00368100 D
11B7  2F 00 01          WR = #0100               ;*                                  00368200 D
11BA  00 1B 11          GOTO GIOEXIT             ;*                                  00368300 D
                GIO1:                                                               00368400 D
11BD  0C 00             CIO = #00                ;*                                  00368500 D
11BF  0C 10             CIO = #10                ;* CLEAR OBAR                       00368600 D
11C1  0C 00             CIO = #00                ;*                                  00368700 D
11C3  0C 01             CIO = #01                ;* PUT GIB IN UNLOAD MODE           00368800 D
11C5  05 00             TX = 0                   ;*                                  00368900 D
11C7  F0                B0 = IO                  ;* READ 256 CHARACTERS              00369000 D
11C8  0C 00             CIO = #00                ;*                                  00369100 D
11CA  0C 10             CIO = #10                ;*                                  00369200 D
11CC  0C 00             CIO = #00                ;*                                  00369300 D
11CE  A2                M2 = M2 - 1              ;*                                  00369400 D
11CF  3A FF             B0 = #FF                 ;*                                  00369500 D
11D1  94                I2 = B0                  ;* SET FLAG                         00369600 D
11D2  2F 00 01          WR = #0100               ;*                                  00369700 D
11D5  00 1B 11          GOTO GIOEXIT             ;*                                  00369800 D
11D8  00        GIOFLAG: PLANT1 0                ;*                                  00369900 D
                END                              ;*                                  00370000 D
                $ POP OMIT                                                          00370100 D
                $ OMIT = NOT HIO                                                    00370200 D
                *                                                                   00370300 D
                *   HOST INPUT-OUTPUT HANDLER                                       00370400 D
                BEGIN                            ;*                                 00370500 D
                LABEL HIOEXIT, HIORDTO, HIOREADIN, HIOREADLOOP, HIOWRITE, HIOWRITELOOP, 00370600 D
                   HIOWRITEOUT, HIOWRTTO, HIOWRTTO1, HIORDTO1, HIONAK, HIOREAD1,    00370700 D
                   HIOWAITR, HIOEXITF, HI01, HI02, HI00;                            00370800 D
                *                                                                   00370900 D
                HIO:                             ;* HIO (BUFFER,SIZE,CODE)          00371000 D
11D9  38 02             AD = #02                 ;* DEVICE ADDRESS                  00371100 D
11DB  F5                B32 FLIP                 ;*                                  00371200 D
11DC  EC                B0 = B2                  ;* SAVE RW                          00371300 D
11DD  31 00             Z = B0 = 0               ;*                                  00371400 D
11DF  25 ** **          IF KS GOTO HI00          ;* KS=>CODE = 0                     00371500 D
11E2  27 ** **          MUTE CALL HI01           ;*                                  00371600 D
11E5  00 ** **          GOTO HIOEXIT             ;*                                  00371700 D
                HI00:                                                               00371800 D
11E8  B7                B1 = IO                  ;* CODE=0.READ FIRST BYTE           00371900 D
11E9  D3                WRL = B1                 ;* MOVE TO WRL                      00372000 D
11EA  AD                WR FLIP                  ;* MOVE TO WRU                      00372100 D
11EB  B7                B1 = IO                  ;* READ SECOND BYTE                 00372200 D
11EC  D3                WRL = B1                 ;* MOVE TO WRL                      00372300 D
                HIOEXIT:                                                            00372400 D
11ED  AD                WR FLIP                  ;*                                  00372500 D
11EE  12 F6 FF          M2 = L - 10              ;*                                  00372600 D
11F1  84                I2 = 0(2)                ;*                                  00372700 D
11F2  00 5E 00          GOTO INTERPF             ;*                                  00372800 D
                HI01:                                                               00372900 D
11F5  BB                K = M1                   ;*                                  00373000 D
11F6  E6                B1 = B0                  ;* CODE                             00373100 D
11F7  17 00             IO = #00                 ;*                                  00373200 D
11F9  1F ** **          IF 3S GOTO J + HI02      ;* 3S=> NAK                         00373300 D
11FC  35 01             B0 = B0 + 1              ;* SET "ACK" BIT                    00373400 D
                HI02:                                                               00373500 D
11FE  EB                CIO = B0                 ;* CONTROL WRITE                    00373600 D
11FF  12 FC FF          M2 = L - 4               ;*                                  00373700 D
```

```
1202  69                    B32 = I2                  ;* READ SIZE                                00373800 D
1203  F5                    B32 FLIP                  ;*                                          00373900 D
1204  CA                    B32 = B32 ! 2             ;* CHANGE WORD COUNT TO HALF WORD COUN00374000 D
1205  2F 0A 00              WR = 10                   ;* TIME LIMIT                                00374100 D
1208  C5                    B1FL FLIP                 ;*                                          00374200 D
                      HIOWAITR:                                                                   00374300 D
1209  B5                    B1 = STAT                 ;*                                          00374400 D
120A  1D ** **              IF 7S GOTO J + HIOREAD1   ;*                                          00374500 D
120D  04 FF FF              WR = WR + #FFFF           ;*                                          00374600 D
1210  25 09 12              IF KS GOTO HIOWAITR       ;*                                          00374700 D
1213  00 ** **              GOTO HIOEXITF             ;*                                          00374800 D
                      HIOREAD1:                                                                   00374900 D
1216  C5                    B1FL FLIP                 ;*                                          00375000 D
1217  1F ** **              IF 3S GOTO J + HIONAK     ;* SUCCESSFUL EXIT FOR NAK                  00375100 D
121A  12 F8 FF              M2 = L - 8                ;* POINTER TO BUFFER ADDRESS                 00375200 D
121D  56                    WR = I2                   ;* READ BUFFER ADDRESS                       00375300 D
121E  AD                    WR FLIP                   ;*                                          00375400 D
121F  A1                    WR = WR ! 2               ;* CHANGE WORD ADDRESS TO BYTE ADDRESS00375500 D
1220  A1                    WR = WR ! 2               ;* CHANGE WORD ADDRESS TO BYTE ADDRESS00375600 D
1221  A4                    M1 = WR                   ;*                                          00375700 D
1222  B1                    B32 = B32 - 1             ;* ADJUST COUNT                              00375800 D
1223  23 ** **              IF 5S GOTO J + HIOWRITE   ;*                                          00375900 D
1226  C5                    B1FL FLIP                 ;*                                          00376000 D
1227  02                    SKIP1                     ;*                                          00376100 D
                      HIOREADLOOP:                                                                00376200 D
1228  B5                    B1 = STAT                 ;* READ STATUS                              00376300 D
1229  1B ** **              IF 6R GOTO J + HIORDTO    ;* 6R => HOST READY                         00376400 D
                      HIOREADIN:                                                                  00376500 D
122C  90                    I1 = IO                   ;* READ DATA                                00376600 D
122D  90                    I1 = IO                   ;* READ DATA                                00376700 D
122E  B1                    B32 = B32 - 1             ;* DECREMENT HALF WORD COUNT                 00376800 D
122F  25 28 12              IF KS GOTO HIOREADLOOP    ;* KS=>MORE DATA                            00376900 D
                      HIONAK:                                                                     00377000 D
1232  3B 54                 WRL = #01                 ;* SET TRUE FLAG                            00377100 D
1234  40                    SKIP2                     ;*                                          00377200 D
                      HIOEXITF:                                                                   00377300 D
1235  3B 55                 WRL = #00                 ;*                                          00377400 D
1237  A0                    WRU = 0                   ;*                                          00377500 D
1238  C2                    M1 = K                    ;*                                          00377600 D
1239  B6                    ENABLE RETURN             ;*                                          00377700 D
                      HIOWRITE:                                                                   00377800 D
123A  C5                    B1FL FLIP                 ;*                                          00377900 D
123B  02                    SKIP1                     ;*                                          00378000 D
                      HIOWRITELOOP:                                                               00378100 D
123C  B5                    B1 = STAT                 ;* READ STATUS                              00378200 D
123D  1B ** **              IF 6R GOTO J + HIOWRTTO   ;* 6R=>HOST NOT READY                       00378300 D
                      HIOWRITEOUT:                                                                00378400 D
1240  5F                    IO = I1                   ;* SEND FIRST BYTE                          00378500 D
1241  5F                    IO = I1                   ;* SEND SECOND BYTE                         00378600 D
1242  B1                    B32 = B32 - 1             ;* DECREMENT HALFWORD COUNT                  00378700 D
1243  25 3C 12              IF KS GOTO HIOWRITELOOP;* KS=>MORE DATA                               00378800 D
1246  00 32 12              GOTO HIONAK               ;*                                          00378900 D
                      HIORDTO:                                                                    00379000 D
1249  2F 0A 00              WR = 10                   ;* SET TIMEOUT COUNT                         00379100 D
                      HIORDTO1:                                                                   00379200 D
124C  B5                    B1 = STAT                 ;* READ STATUS                              00379300 D
124D  22 2C 12              IF 6S GOTO J + HIOREADIN  ;* 6S=>HOST READY                          00379400 D
1250  04 FF FF              WR = WR + #FFFF           ;* DECREMENT COUNTER                        00379500 D
1253  25 4C 12              IF KS GOTO HIORDTO1       ;*                                          00379600 D
1256  00 35 12              GOTO HIOEXITF             ;*                                          00379700 D
```

```
                        HIOWRTTO:                                                                   00379800 D
1259  2F 0A 00                  WR = 10                     ;* SET TIMEOUT COUNT                  00379900 D
                        HIOWRTTO1:                                                                  00380000 D
125C  B5                        B1 = STAT                   ;* READ STATUS                         00380100 D
125D  22 40 12                  IF 6S GOTO J + HIOWRITEOUT  ;* 6S=>HOST READY                     00380200 D
1260  04 FF FF                  WR = WR + #FFFF             ;* DECREMENT COUNTER                   00380300 D
1263  25 5C 12                  IF KS GOTO HIOWRTTO1        ;*                                   00380400 D
1266  00 35 12                  GOTO HIOEXITF               ;*                                   00380500 D
                        END                                 ;*                                   00380600 D
                        $ POP OMIT                                                                  00380700 D
                        $ OMIT = NOT CRT                                                            00380800 D
                        *        NEW CIO OPERATOR                                                   00380900 D
                        *                                                                           00381000 D
                        BEGIN                               ;*                                      00381100 D
                        LABEL ACIACMND = #81, ACIASTAT = #C1, ACIAREAD = #C0, ACIAWRITE = #80,      00381200 D
                        $ OMIT = G13                                                                00381210
                        $ POP OMIT + OMIT = NOT G13                                                 00381310
                         ENABLEREAD = #92, ENABLEWRITE = #32, ACIARESET = #03, BAUD9600 = 0,        00381320
                        $ POP OMIT                                                                  00381330
                           ASYNCHMASK = #FD, ASYNCHCHANNEL = #02, ENQ=#05, ACK=#06, STX=#02,        00381400 D
                        $ OMIT = G13                                                                00381410
                        $ POP OMIT + OMIT = NOT G13                                                 00381510
                         READ4,SNDRESP,READDATA,READ3,READ2,EXITTRUE,EXITFALSE,                     00381520
                        $ POP OMIT                                                                  00381530
                           WRITE, WRITEDATA, WRITE1, WRITE2, WRITE3, SENDBCC, CRTREAD, CIOFLAG,     00381600 D
                           CRTWRITE, CRTWRITE1, CRTWRITE2, CRTREAD1, CRTREAD2, READRELOC,           00381700 D
                        $ OMIT = G13                                                                00381710
                        $ POP OMIT + OMIT = NOT G13                                                 00381810
                        NAK=#15,SOH=#81,EOT=#84,DELAY,WRITE4,CIO1,TURNAROUND;                       00381820
                        $ POP OMIT                                                                  00381830
                        *                                                                           00381900 D
                        CIO:                                * CIO (UNIT, BUFFER, SIZE)              00382000 D
1269  07 ** **                  M2 = CIOFLAG                ;*                                    00382100 D
126C  53                        B0 = I2                     ;*                                    00382200 D
126D  26 ** **                  IF KR GOTO CIO1             ;* TEST FOR FIRST CALL OF CIO          00382300 D
1270  12 F9 FF                  M2 = L - 7                  ;*                                    00382400 D
1273  53                        B0 = I2                     ;*                                    00382500 D
                        $ OMIT = NOT OLDLOOP                                                        00382600 D
                        $ POP OMIT                                                                  00382900 D
1274  36 02                     AD = ASYNCHCHANNEL          ;*                                    00383000 D
1276  BB                        K = M1                      ;*                                    00383100 D
1277  56                        WR = I2                     ;*                                    00383200 D
1278  56                        WR = I2                     ;*                                    00383300 D
1279  AD                        WR FLIP                     ;*                                    00383400 D
127A  A1                        WR = WR ! 2                 ;*                                    00383500 D
127B  A1                        WR = WR ! 2                 ;*                                    00383600 D
127C  54                        B1 = I2                     ;*                                    00383700 D
127D  A4                        M1 = WR                     ;*                                    00383800 D
127E  F5                        B32 FLIP                    ;*                                    00383900 D
127F  CA                        B32 = B32 ! 2               ;*                                    00384000 D
1280  CA                        B32 = B32 ! 2               ;*                                    00384100 D
                        $ OMIT = NOT G13                                                            00384110
1281  3A 06                     B0 = 6                      ;* COUNT FOR HEADER                    00384120
1283  CC                        XU = B0                     ;* SAVE COUNT                           00384130
                        $ POP OMIT                                                                  00384140
1284  22 ** **                  IF 6S GOTO J + WRITE        ;*                                    00384200 D
1287  B1                        B32 = B32 - 1               ;*                                    00384300 D
1288  19 ** **                  IF KR GOTO J + EXITTRUE     ;*                                    00384400 D
128B  F3                        B0 = REQ                    ;*                                    00384500 D
128C  3E FD                     Z = B0 ▼ ASYNCHMASK         ;*                                    00384600 D
```

```
128E  19 ** **          IF KR GOTO J + EXITFALSE  ;*                                    00384700 D
                    $ OMIT = G13                                                        00384710
                    $ POP OMIT                                                          00385610
                    READDATA:                                                           00385700 D
1291  1E ** **          CALL J + CRTREAD          ;*                                    00385800 D
                    $ OMIT = NOT G13                                                    00385805
1294  A5                WRL = B0                  ;*                                    00385810
1295  C5                B1FL FLIP                 ;*                                    00385815
1296  E5                B0 = B0 \ WRL             ;*                                    00385820
1297  E6                B1 = B0                   ;*                                    00385825
1298  C5                B1FL FLIP                 ;* SAVE BCC                           00385830
1299  CD                B0 = XU                   ;* FETCH HEADER COUNT                  00385835
129A  31 00             Z = B0 = #00              ;* TEST FOR ZERO                       00385840
129C  26 ** **          IF KR GOTO READ4          ;* KR => STILL IN HEADER              00385845
129F  DB                B0 = WRL                  ;*                                    00385850
                    $ POP OMIT                                                          00385855
12A0  02 E0             Z = B0 - #20              ;*                                    00385900 D
12A2  19 ** **          IF KR GOTO J + READRELOC  ;* ADD #40                            00386000 D
12A5  02 A0             Z = B0 - #60              ;*                                    00386100 D
12A7  18 ** **          IF KS GOTO J + READ3      ;*                                    00386200 D
12AA  35 E0             B0 = B0 - #20             ;*                                    00386300 D
12AC  A5                WRL = B0                  ;*                                    00386400 D
12AD  A0                WRU = 0                   ;*                                    00386500 D
12AE  06 35 0E          M2 = WRR + ASCINT         ;*                                    00386600 D
12B1  53                B0 = I2                   ;*                                    00386700 D
12B2  40                SKIP2                     ;*                                    00386800 D
                    READRELOC:                                                          00386900 D
12B3  35 40             B0 = B0 + #40             ;*                                    00387000 D
                    READ3:                                                              00387100 D
12B5  B1                B32 = B32 - 1             ;*                                    00387200 D
12B6  18 ** **          IF KS GOTO J + READ2      ;*                                    00387300 D
12B9  3C 00 00          B32 = 0                   ;*                                    00387400 D
12BC  02                SKIP1                     ;*                                    00387500 D
                    READ2:                                                              00387600 D
12BD  89                I1 = B0                   ;*                                    00387700 D
12BE  31 43             Z = B0 = #43              ;*                                    00387800 D
12C0  19 91 12          IF KR GOTO J + READDATA   ;*                                    00387900 D
12C3  2E ** **          CALL CRTREAD              ;* GET BCC                            00388000 D
                    $ OMIT = NOT G13                                                    00388010
12C6  00 ** **          GOTO EXITTRUE              ;*                                   00388020
12C9  C5                B1FL FLIP                 ;*                                    00388030
12CA  DC                B0 = B1                   ;*                                    00388040
12CB  E5                B0 = B0 \ WRL             ;*                                    00388050
                    $ POP OMIT                                                          00388060
12CC  2E ** **          CALL TURNAROUND           ;* DELAY 2 MS                        00388100 D
12CF  37 B1             B1 = ACK                  ;*                                    00388200 D
                    $ OMIT = NOT G13                                                    00388210
12D1  31 00             Z = B0 = #00              ;*                                    00388220
12D3  00 ** **               GOTO SNDRESP         ;* KS => BCC OK                      00388230
12D6  37 C0             B1 = NAK                  ;*                                    00388240
                    SNDRESP:                                                            00388250
                    $ POP OMIT                                                          00388260
12D8  2E ** **          CALL CRTWRITE             ;*                                    00388300 D
                    $ OMIT = G13                  ;*                                    00388310
                    $ POP OMIT                                                          00388510
                    EXITTRUE:                                                           00388600 D
12DB  3B 54             WRL = #01                 ;*                                    00388700 D
12DD  40                SKIP2                     ;*                                    00388800 D
                    EXITFALSE:                                                          00388900 D
12DE  3B 55             WRL = #00                 ;*                                    00389000 D
```

```
12E0  A0                WRU = 0                 ;*                              00389100 D
12E1  AD                WR FLIP                 ;*                              00389200 D
12E2  0C 81             CIC = ACIACMND          ;*                              00389300 D
12E4  17 92             IO = ENABLEREAD         ;* ALLOW INTERRUPT              00389400 D
12E6  C2                M1 = K                  ;*                              00389500 D
12E7  12 F6 FF          M2 = L - 10             ;*                              00389600 D
12EA  84                I2 = 0(2)               ;*                              00389700 D
12EB  00 5E 00          GOTO INTERPF            ;*                              00389800 D
                  $ OMIT=NOT G13                                                  00389810
                  READ4:                                                        00389820
12EE  35 FF             B0 = B0 - 1             ;* DECREMENT HEADER COUNT        00389830
12F0  CC                XU = B0                 ;* SAVE COUNT                    00389840
12F1  DB                B0 = WRL                ;*                              00389850
12F2  00 B5 12          GOTO READ3              ;*                              00389860
                  $ POP OMIT                                                    00389870
                  WRITE:                                                        00389900 D
                  $ OMIT = G13                                                  00389910
                  $ POP OMIT + OMIT = NOT G13                                   00391010
12F5  37 AB             B1 = #00                ;*                              00391020
                  $ POP OMIT                                                    00391030
12F7  C5                B1FL FLIP               ;*                              00391100 D
                  WRITEDATA:                                                    00391200 D
12F8  B1                B32 = B32 - 1           ;*                              00391300 D
12F9  18 ** **          IF KS GOTO J + WRITE1   ;*                              00391400 D
12FC  3A 43             B0 = #43                ;*                              00391500 D
12FE  02                SKIP1                   ;*                              00391600 D
                  WRITE1:                                                       00391700 D
12FF  64                B0 = I1                 ;*                              00391800 D
                  $ OMIT = NOT G13                                              00391810
1300  E6                B1 = B0                 ;* SAVE CHAR                     00391820
1301  CD                B0 = XU                 ;*                              00391830
1302  31 00             Z = B0 = #00            ;*                              00391840
1304  26 ** **          IF KR GOTO WRITE4       ;*                              00391850
1307  DC                B0 = B1                 ;*                              00391860
                  $ POP OMIT                                                    00391870
1308  02 A0             Z = B0 - #60            ;*                              00391900 D
130A  18 ** **          IF KS GOTO J + WRITE3   ;* NO CHANGE                     00392000 D
130D  02 C0             Z = B0 - #40            ;*                              00392100 D
130F  18 ** **          IF KS GOTO J + WRITE2   ;* RELOC                         00392200 D
1312  A5                WRL = B0                ;*                              00392300 D
1313  A0                WRU = 0                 ;*                              00392400 D
1314  06 F5 0D          M2 = WRR + INTASC       ;*                              00392500 D
1317  53                B0 = I2                 ;*                              00392600 D
1318  40                SKIP2                   ;*                              00392700 D
                  WRITE2:                                                       00392800 D
1319  35 C0             B0 = B0 - #40           ;*                              00392900 D
                  $ OMIT = G13                                                  00392910
                  $ POP OMIT                                                    00393110
                  WRITE3:                                                       00393200 D
131B  A5                WRL = B0                ;*                              00393300 D
131C  C5                B1FL FLIP               ;*                              00393400 D
131D  DC                B0 = B1                 ;*                              00393500 D
131E  E5                B0 = B0 \ WRL           ;*                              00393600 D
131F  E6                B1 = B0                 ;*                              00393700 D
1320  C5                B1FL FLIP               ;*                              00393800 D
1321  BE                B1 = WRL                ;*                              00393900 D
1322  1E ** **          CALL J + CRTWRITE       ;*                              00394000 D
1325  DC                B0 = B1                 ;*                              00394100 D
1326  31 03             Z = B0 = #03            ;*                              00394200 D
1328  19 F8 12          IF KR GOTO J + WRITEDATA ;*                             00394300 D
```

```
                        SENDBCC:                                                    00394400 D
                        $ OMIT = G13                                                00394410
                        $ POP OMIT                                                  00394805
                        $ OMIT = G13                                                00394810
                        $ POP OMIT                                                  00395110
                        $ OMIT = G13                                                00395120
                        $ POP OMIT                                                  00395510
132B  1C DB 12                GOTO J + EXITTRUE             ;*                      00395600 D
                        WRITE4:                                                     00395700 D
                        $ OMIT = G13                                                00395710
                        $ POP OMIT + OMIT = NOT G13                                 00396610
132E  35 FF                   B0 = B0 - 1                   ;*                      00396620
1330  CC                      XU = B0                       ;*                      00396630
1331  DC                      B0 = B1                       ;*                      00396640
                        $ POP OMIT                                                  00396650
1332  00 1B 13                GOTO WRITE3                   ;*                      00396700 D
                        TURNAROUND:                                                 00396800 D
1335  37 AB                   B1 = 0                        ;*                      00396900 D
1337  42 01                   B1 = B1 + 1                   ;*                      00397000 D
1339  26 37 13                IF KR GOTO TURNAROUND + 2     ;*                      00397100 D
                        $ OMIT = G13                                                00397110
                        $ POP OMIT                                                  00397410
133C  A6                      RETURN                        ;*                      00397500 D
                        CRTREAD:                                                    00397600 D
133D  0C 81                   CIO = ACIACMND                ;*                      00397700 D
133F  17 92                   IO = ENABLEREAD               ;*                      00397800 D
1341  2F 00 80                WR = #8000                    ;*                      00397900 D
                                                                                    00398000 D
                        CRTREAD1:                                                   00398100 D
1344  F3                      B0 = REQ                      ;*                      00398200 D
1345  3E FD                   Z = B0 V ASYNCHMASK           ;*                      00398300 D
1347  18 ** **                IF KS GOTO J + CRTREAD2       ;*                      00398400 D
134A  04 FF FF                WR = WR + #FFFF               ;*                      00398500 D
134D  25 44 13                IF KS GOTO CRTREAD1           ;*                      00398600 D
1350  00 DE 12                GOTO EXITFALSE                ;* TIMEOUT             00398700 D
                        CRTREAD2:                                                   00398800 D
1353  0C C0                   CIO = ACIAREAD                ;*                      00398900 D
1355  D7                      NOOP                          ;*                      00399000 D
1356  F0                      B0 = IO                       ;*                      00399100 D
1357  A6                      RETURN                        ;*                      00399200 D
                        CRTWRITE:                                                   00399300 D
1358  0C 81                   CIO = ACIACMND                ;*                      00399400 D
135A  17 32                   IO = ENABLEWRITE              ;*                      00399500 D
135C  2F 00 01                WR = #0100                    ;*                      00399600 D
                        CRTWRITE1:                                                  00399700 D
135F  F3                      B0 = REQ                      ;*                      00399800 D
1360  2B FD                   B0 = B0 V ASYNCHMASK          ;*                      00399900 D
1362  18 ** **                IF KS GOTO J + CRTWRITE2      ;*                      00400000 D
1365  04 FF FF                WR = WR + #FFFF               ;*                      00400100 D
1368  25 5F 13                IF KS GOTO CRTWRITE1          ;*                      00400200 D
136B  00 DE 12                GOTO EXITFALSE                ;* TIMEOUT             00400300 D
                        CRTWRITE2:                                                  00400400 D
136E  0C 80                   CIO = ACIAWRITE               ;*                      00400500 D
1370  DC                      B0 = B1                       ;*                      00400600 D
1371  EA                      IO = B0                       ;*                      00400700 D
1372  A6                      RETURN                        ;*                      00400800 D
                        DELAY:                                                      00400900 D
1373  2E 35 13                CALL TURNAROUND               ;* DELAY 2 MS           00401000 D
1376  00 DE 12                GOTO EXITFALSE                ;*                      00401100 D
1379  00                CIOFLAG: PLANT1 0                   ;*                      00401200 D
```

```
                        CIO1:                                                                                   00401300 D
137A  38 02                     AD = #02                ;* INITIALIZE ASYNCH INTERFACE                          00401400 D
137C  0C 00                     CIO = #00               ;* SET BAUD RATE TO 9600                                 00401500 D
137E  0C 81                     CIO = #81               ;* ACIACMND                                             00401600 D
1380  17 03                     IO = #03                ;* ACIARESET                                            00401700 D
1382  0C 81                     CIO = ACIACMND          ;*                                                     00401800 D
1384  17 92                     IO = ENABLEREAD         ;*                                                     00401900 D
1386  A2                        M2 = M2 - 1             ;*                                                     00402000 D
1387  3A FF                     B0 = #FF                ;*                                                     00402100 D
1389  94                        I2 = B0                 ;* SET FLAG                                             00402200 D
138A  BB                        K = M1                  ;*                                                     00402300 D
138B  00 DB 12                  GOTO EXITTRUE           ;*                                                     00402400 D
                        END     ;                                                                               00402500 D
                        $ POP OMIT                                                                              00402600 D
                        $OMIT = NOT PIO                                                                         00402700 D
                        $POP OMIT + OMIT = NOT TCCF                                                             00425000 D
                        *       TCCF INTERFACE OPERATOR (BDS SYNCHRONOUS DC CONTROLLER)                         00425100 D
                        *                                                                                       00425200 D
                        BEGIN                           ;*                                                      00425300 D
                        LABEL SYNCHCHAR = #22, FILLER = #55, USRTREAD = #00, USRTSTATUS = #01,                  00425400 D
                           USRTRESET = #02, USRTRESTART = #03, USRTWRITE = #04,                                 00425500 D
                           USRTLDSYNCH = #05, USRTLDFILL = #06, USRTLDFORM = #07, NOMODEM = #80,                00425600 D
                           SLUSRTRDA = #C0, SLUSRTTBMT = #C1, SLUSRTSCR = #C2, RECEIVE,                         00425700 D
                           TRANSMIT, TRANSMIT1, EXITTRUE, EXITFALSE, READLOOP, WRITELOOP,                       00425800 D
                           CHECKSTATUS, CHECKSTATUS1, TCCFLAG, TCCF1, MEMDATA;                                  00425900 D
                        *                                                                                       00426000 D
                        *                                                                                       00426100 D
                        *                                                                                       00426200 D
                        *                                                                                       00426300 D
                        TCCF:                           * TCCF ( BUFFER, SIZE)                                  00426400 D
138E  12 00 00                  M2 = L + 0              ;*                                                     00426500 D
1391  83                        I2=B32                  ;*                                                     00426600 D
1392  07 ** **                  M2 = TCCFLAG            ;*                                                     00426700 D
1395  53                        B0 = I2                 ;* READ FLAG                                           00426800 D
1396  26 ** **                  IF KR GOTO TCCF1        ;* IF FLAG NOT SET,GOTO INITIALIZE                      00426900 D
1399  38 02                     AD = #02                ;*                                                     00427000 D
139B  12 FC FF                  M2 = L - 4              ;* POINT TO BUFFER ADDRESS                               00427100 D
139E  56                        WR = I2                 ;*                                                     00427200 D
139F  AD                        WR FLIP                 ;*                                                     00427300 D
13A0  A1                        WR = WR ! 2             ;* MAKE BYTE ADDRESS                                     00427400 D
13A1  A1                        WR = WR ! 2             ;*MAKE BYTE ADDRESS                                    00427500 D
13A2  54                        B1 = I2                 ;* READ SIGN OF SIZE                                   00427600 D
13A3  F5                        B32 FLIP                ;*                                                     00427700 D
13A4  BB                        K = M1                  ;*                                                     00427800 D
13A5  A4                        M1 = WR                 ;*                                                     00427900 D
13A6  3A 01                     B0 = #01                ;*                                                     00428000 D
13A8  B1                        B32 = B32 - 1           ;* ADJUST SIZE                                          00428100 D
13A9  26 ** **                  IF KR GOTO EXITTRUE     ;*                                                     00428200 D
13AC  22 ** **                  IF 6S GOTO J + TRANSMIT ;*                                                     00428300 D
13AF  0C C0                     CIO = SLUSRTRDA         ;*                                                     00428400 D
                        *                                                                                       00428500 D
                        RECEIVE:                                                                                00428600 D
13B1  2F 00 03                  WR = #300               ;* SET COUNT                                            00428700 D
                        READLOOP:                                                                               00428800 D
13B4  F3                        B0 = REQ                ;*                                                     00428900 D
13B5  3E FD                     Z = B0 V #FD            ;* TEST FOR INTERRUPT ON CHANNEL 2                      00429000 D
13B7  25 ** **                  IF KS GOTO CHECKSTATUS  ;*                                                     00429100 D
13BA  04 FF FF                  WR = WR - 1             ;*                                                     00429200 D
13BD  25 B4 13                  IF KS GOTO READLOOP     ;*                                                     00429300 D
                        EXITFALSE:                                                                              00429400 D
```

```
13C0  0C 03              CIC = USRTRESTART          ;*                                  00429500 D
13C2  EA                 IO = B0                    ;*                                  00429600 D
13C3  0C C2              CIO = SLUSRTSCR            ;*                                  00429700 D
13C5  3A 00              B0 = 0                     ;*                                  00429800 D
                    *                                                                   00429900 D
                    EXITTRUE:                                                           00430000 D
13C7  12 00 00           M2 = L + 0                 ;*                                  00430100 D
13CA  56                 WR = I2                    ;* READ SIZE                        00430200 D
13CB  AD                 WR FLIP                    ;*                                  00430300 D
13CC  C7                 WR = WR - B32              ;* SUBTRACT REMAINING COUNT          00430400 D
13CD  04 FF FF           WR = WR - 1                ;*                                  00430500 D
13D0  C2                 M1 = K                     ;*                                  00430600 D
13D1  12 FA FF           M2 = L - 6                 ;* ADJUST TOS                        00430700 D
13D4  05 01              TX = 1                     ;*                                  00430800 D
13D6  84                 I2 = 0(2)                  ;*                                  00430900 D
13D7  AD                 WR FLIP                    ;*                                  00431000 D
13D8  82                 I2 = WR                    ;*                                  00431100 D
13D9  94                 I2 = B0                    ;*                                  00431200 D
13DA  00 53 00           GOTO INTERPE               ;*                                  00431300 D
                    CHECKSTATUS:                                                        00431400 D
13DD  0C 01              CIO = USRTSTATUS           ;*                                  00431500 D
13DF  B7                 B1 = IO                    ;* READ USRT STATUS                 00431600 D
13E0  20 B1 13           IF 0R GOTO J + RECEIVE     ;* 0R => RDA NOT SET                00431700 D
13E3  0C 00              CIO = USRTREAD             ;*                                  00431800 D
13E5  90                 I1 = IO                    ;* READ CHARACTER                   00431900 D
13E6  B1                 B32 = B32 - 1              ;*                                  00432000 D
13E7  25 B1 13           IF KS GOTO RECEIVE         ;*                                  00432100 D
13EA  0C 03              CIO = USRTRESTART          ;* PLACE USRT IN SEARCH MODE         00432200 D
13EC  EA                 IO = B0                    ;* DUMMY I/O INSTRUCTION             00432300 D
13ED  0C C2              CIO = SLUSRTSCR            ;* SELECT SRC AS INTERRUPT SOURCE    00432400 D
13EF  3A 01              B0 = #01                   ;* SET TRUE FLAG                     00432500 D
13F1  00 C7 13           GOTO EXITTRUE              ;*                                  00432600 D
                    *                                                                   00432700 D
                    TRANSMIT:                                                           00432800 D
13F4  37 2B              B1 = #80                   ;* SET SYNC FLAG                     00432900 D
13F6  0C C1              CIO = SLUSRTTBMT           ;* SELECT TBMT AS INTERRUPT SOURCE   00433000 D
13F8  C5                 B1FL FLIP                  ;*                                  00433100 D
                    TRANSMIT1:                                                          00433200 D
13F9  2F 00 03           WR = #300                  ;*                                  00433300 D
                    WRITELOOP:                                                          00433400 D
13FC  F3                 B0 = REQ                   ;*                                  00433500 D
13FD  3E FD              Z = B0 V #FD               ;*                                  00433600 D
13FF  25 ** **           IF KS GOTO CHECKSTATUS1    ;*                                  00433700 D
1402  04 FF FF           WR = WR - 1                ;*                                  00433800 D
1405  25 FC 13           IF KS GOTO WRITELOOP       ;*                                  00433900 D
1408  00 C0 13           GOTO EXITFALSE             ;*                                  00434000 D
                    CHECKSTATUS1:                                                       00434100 D
140B  0C 01              CIC = USRTSTATUS           ;* READ USRT STATUS                 00434200 D
140D  B7                 B1 = IO                    ;*                                  00434300 D
140E  1A F9 13           IF 7R GOTO J + TRANSMIT1   ;* 7R => TBMT NOT SET              00434400 D
1411  C5                 B1FL FLIP                  ;*                                  00434500 D
1412  1A ** **           IF 7R GOTO J + MEMDATA     ;* 7R => SYNCH CHAR ALREADY SENT    00434600 D
1417  C5 AB              B1 = #00;B1FL FLIP         ;*                                  00434700 D
1418  0C 04              CIC = USRTWRITE            ;*                                  00434800 D
141A  17 22              IO = SYNCHCHAR             ;* TRANSMIT SYNCH CHAR              00434900 D
141F  00 F9 13           B32 = B32 + 1;GOTO TRANSMIT1 ;* ADJUST COUNT                   00435000 D
                    MEMDATA:                                                            00435100 D
1422  0C 04              CIO = USRTWRITE            ;* WRITE CHARACTER TO USRT           00435200 D
1425  C5                 IO = I1;B1FL FLIP          ;*                                  00435300 D
1426  B1                 B32 = B32 - 1              ;*                                  00435400 D
```

```
1427  25 F9 13            IF KS GOTO TRANSMIT1        ;*                                  00435500 D
142A  0C 03               CIO = USRTRESTART           ;* PLACE USRT IN SEARCHMODE          00435600 D
142C  EA                  IO = B0                     ;* DUMMY I/O INTRUCTION               00435700 D
142D  0C C2               CIO = SLUSRTSCR             ;* SELECT SRC AS INTERRUPT SOURCE     00435800 D
142F  3A 01               B0 = #01                    ;*                                  00435900 D
1431  00 C7 13            GOTO EXITTRUE               ;*                                  00436000 D
                    *                                                                     00436100 D
                    TCCF1:                                                                00436200 D
1434  38 02               AD = #02                    ;*                                  00436300 D
1436  0C 02               CIO = USRTRESET             ;* INITIALIZE USRT                    00436400 D
1438  EA                  IO = B0                     ;* DUMMY I/O INSTRUCTION              00436500 D
1439  D7                  NOOP                        ;* DELAY                             00436600 D
143A  D7                  NOOP                        ;*                                  00436700 D
143B  D7                  NOOP                        ;*                                  00436800 D
143C  0C 07               CIO = USRTLDFORM            ;* LOAD CHARACTER FORMAT             00436900 D
143E  17 0B               IO = #0B                    ;* 8 BIT CHARACTERS, NO PARITY        00437000 D
1440  0C 05               CIO = USRTLDSYNCH           ;* LOAD SYNC CHARACTER               00437100 D
1442  17 22               IO = SYNCHCHAR              ;*                                  00437200 D
1444  0C 06               CIO = USRTLDFILL            ;* LOAD TRANSMIT FILLER CHARACTER    00437300 D
1446  17 55               IO = FILLER                 ;*                                  00437400 D
1448  0C 03               CIO = USRTRESTART           ;*                                  00437500 D
144A  B7                  B1 = IO                     ;* DUMMY I/O                         00437600 D
144B  0C C2               CIO = SLUSRTSCR             ;* SELECT SCR AS INTERRUPT SOURCE     00437700 D
144D  A2                  M2 = M2 - 1                 ;* BACK UP MEMORY POINTER             00437800 D
144E  3A FF               B0 = #FF                    ;*                                  00437900 D
1450  94                  I2 = B0                     ;* SET FLAG                           00438000 D
1451  3A 01               B0 = #01                    ;*                                  00438100 D
1453  00 C7 13            GOTO EXITTRUE               ;*                                  00438200 D
1456  00                  TCCFLAG: PLANT1 0           ;*                                  00438300 D
                    END                               ;*                                  00438400 D
                    $POP OMIT + OMIT = NOT SDLC                                           00438500 D
                    $POP OMIT                                                             00452600 D
                    OPTABLE:                                                              00452700 D
1457  9C 0E 41 02         OP8  EROP,   ADD,   SUB,   MUL,  EROP,   MOD,   DIV,  EROP ;    00452800 D
145B  36 02 5C 03
145F  9C 0E 2F 04
1463  2F 04 9C 0E
1467  9C 0E BB 06         OP8  EROP,  EQVL,  NEQV,  GRTR,  LEQV,  GEQV,  LSTR,  BAND ;    00452900 D
146B  B8 06 60 06
146F  3F 06 56 06
1473  4C 06 07 01
1477  24 01 41 01         OP8   BOR,  BEQV,   SCN,   DLT,   XCH,   NDX,   STN,   STD ;    00453000 D
147B  6F 0B DC 0A
147F  23 02 6D 06
1483  6A 01 6A 01
1487  9F 01 AC 01         OP8   CBF,   CBB,   CIO,  CADD,  CSUB,   CHR,  XINT,   TBL ;    00453100 D
148B  69 12 BC 08
148F  B4 08 BA 0A
1493  3A 0C D5 0B
1497  F3 00 ED 01         OP8  BNOT,   CHS,   SSP,ENABLE,   LOT,   STS,   GTS,  SETP ;    00453200 D
149B  0D 02 1A 02
149F  57 08 C9 01
14A3  DD 06 92 0E
14A7  9C 0E AB 06         OP8  EROP,   LOD,   DUP,   ENT,   RND,   BRB,   BRF,  EVAL ;    00453300 D
14AB  B9 01 A4 06
14AF  A4 06 94 01
14B3  8A 01 86 07
14B7  F9 02 1F 03         OP8  ADD1,  SUB1, ZEROX,  NEG1,  PORV,  UNLK,  LOCK,   RDY ;    00453400 D
14BB  D3 01 E7 00
14BF  9B 06 9B 06
```

```
14C3  9B 06 97 06
14C7  9C 0E E5 07          OP8   RTR,   RTN,   XIT,   NTR,   L16,   DEL,   NOP,  HALT ;    00453500 D
14CB  E9 07 96 07
14CF  DE 00 C3 01
14D3  A4 06 75 0E
14D7  53 0D 0D 09          OP8   XTR,   TRC,   TRW,   TRS,   TRD,   TRB,   TRH,   DIO ;    00453600 D
14DB  09 09 D4 08
14DF  82 09 BC 0C
14E3  74 0A 9C 0E
14E7  F9 0B 9C 0E          OP8   CEQ,  EROP,  EROP,  EROP,  EROP,  EROP,  EROP,  EROP ;    00453700 D
14EB  9C 0E 9C 0E
14EF  9C 0E 9C 0E
14F3  9C 0E 9C 0E
14F7  9C 0E 9C 0E          OP8  EROP,  EROP,   SMX,   ZIP,  EROP,   PUT,   GET,   LDM ;    00453800 D
14FB  9C 0E 9C 0E
14FF  9C 0E 35 07
1503  22 07 9C 0E
1507  9C 0E 9C 0E          OP8   WRT,   RED,  HASH,   TRX,  TRX1, SCAN1,  EROP,   PCH ;    00453900 D
150B  35 08 FF 0A
150F  41 0B 92 0B
1513  9C 0E 9C 0E
1517  8C 0F 2E 10          OP8   LIO,   RAM,STATUS,   GIO,  TIME,  WTKN,   HIO, PANEL ;    00454000 D
151B  62 10 F3 10
151F  90 10 A2 10
1523  D9 11 9C 0E
1527  9C 0E 83 10          OP8   PIO, SWTCH,  TCCF,  SDLC,  EROP,  EROP,  EROP,  EROP ;    00454100 D
152B  8E 13 9C 0E
152F  9C 0E 9C 0E
1533  9C 0E 9C 0E
1537  9C 0E 9C 0E          OP8  EROP,  EROP,  EROP,  EROP,  EROP,  EROP,  EROP,  EROP ;    00454200 D
153B  9C 0E 9C 0E
153F  9C 0E 9C 0E
1543  9C 0E 9C 0E
1547  9C 0E 9C 0E          OP8  EROP,  EROP,  EROP,  EROP,  EROP,  EROP,  EROP,  EROP ;    00454300 D
154B  9C 0E 9C 0E
154F  9C 0E 9C 0E
1553  9C 0E 9C 0E                                                                         00454400 D

                    $ OMIT = NOT TRACE                                                  00454500 D
                    $OMIT = NOT TRACE                                                   00454600 D
                    $ OMIT = NOT HIO                                                    00458100 D
                    $ POP OMIT                                                          00460400 D
                    $ POP OMIT                                                          00461300 D
                    $ POP OMIT                                                          00461400 D
                    $ LIST CLIST CODE                                                   00461500 D
                                                                                        00461600 D
                    STACKBASE:4: END                ;* ALIGN TO 4-BYTE BOUNDARY         00461700 D
                    $ CODE                                                              00461800 D
 0 ERRORS   4133 CARDS.  10597 TOKENS.     20294 RULES.       474 SECONDS.             00461900 D

1558  A5 A5 A5 A5
 31     DISK SECTORS
1414 14   1413 22   1401 14   1400 0B   13B9 13   13B8 DD   13AE 13   13AD F4   13AB 13   13AA C7   1398 14   1397 34   1394 14
1393 56   1364 13   1363 6E   1349 13   1348 53   1324 13   1323 58   1311 13   1310 19   130C 13   130B 1B   1306 13   1305 2E
12FB 12   12FA FF   12DA 13   12D9 58   12D5 12   12D4 D8   12CE 13   12CD 35   12C8 12   12C7 DB   12C5 13   12C4 3D   12B8 12
12B7 BD   12A9 12   12A8 B5   12A4 12   12A3 B3   129E 12   129D EE   1293 13   1292 3D   1290 12   128F DE   128A 12   1289 DB
1286 12   1285 F5   126F 13   126E 7A   126B 13   126A 79   123F 12   123E 59   122B 12   122A 49   1225 12   1224 3A   1219 12
1218 32   1215 12   1214 35   120C 12   120B 16   11FB 11   11FA FE   11E7 11   11E6 ED   11E4 11   11E3 F5   11E1 11   11E0 E8
1191 11   1190 99   118C 11   118B 9B   1186 11   1185 A3   116D 11   116C 71   1142 11   1141 4E   113D 11   113C 4C   1136 11
1135 4E   1117 11   1116 22   110E 11   110D 64   10F9 11   10F8 BD   10F5 11   10F4 D8   10F0 10   10EF F2   10D7 10   10D6 E2
```

10A9 10	10A8 B4	1069 10	1068 79	104E 10	104D 58	104B 10	104A 4F	1046 10	1045 4F	1041 10	1040 4F	103C 10
103B 4F	101D 10	101C 21	1002 10	1001 07	0FFA 0F	0FF9 FE	0FEF 0F	0FEE F3	0FE7 10	0FE6 10	0FE2 0F	0FE1 E4
0FDC 0F	0FDB DF	0FD1 0F	0FD0 D7	0FC9 0F	0FC8 D2	0FC0 0F	0FBF C3	0FB9 0F	0FB8 BC	0FB4 10	0FB3 16	0FAD 0F
0FAC B0	0F70 15	0F6F E4	0F2B 0F	0F2A 41	0F1D 0F	0F1C 48	0F17 0F	0F16 55	0F0F 0F	0F0E 56	0F06 0F	0F05 46
0EFB 0F	0EFA 01	0EF8 0F	0EF7 1B	0EF5 0E	0EF4 FC	0EF2 0F	0EF1 26	0EE5 0F	0EE4 48	0ED2 15	0ED1 D8	0EC6 0E
0EC5 C9	0EBF 0E	0EBE C9	0EB3 0E	0EB2 C9	0EAE 0E	0EAD C9	0EA0 0E	0E9F C9	0E8E 15	0E8D D8	0E85 15	0E84 58
0E77 15	0E76 5A	0DC8 0D	0DC7 CC	0DC1 0D	0DC0 C5	0DBA 0D	0DB9 BE	0DB0 0D	0DAF ED	0DAB 0D	0DAA E9	0D84 0D
0D83 9C	0D7B 0D	0D7A 7E	0D75 0D	0D74 A7	0D44 0D	0D43 52	0D39 0D	0D38 3D	0D32 0D	0D31 3A	0CFF 0D	0CFE 29
0CF7 0C	0CF6 FC	0CED 0C	0CEC F8	0CD7 0D	0CD6 29	0CB8 0C	0CB7 BB	0CA2 0C	0CA1 A5	0C8C 0C	0C8B 90	0C7D 0C
0C7C 81	0C75 0C	0C74 79	0C6E 0C	0C6D 72	0C63 0C	0C62 67	0C5D 0C	0C5C 96	0C4D 0C	0C4C 51	0C27 0C	0C26 2B
0C20 0D	0C1F F5	0C19 0D	0C18 F5	0C0D 0C	0C0C 2E	0C07 0C	0C06 2E	0BF0 0B	0BEF F3	0BB6 0B	0BB5 C9	0B9F 0B
0B9E B7	0B91 0B	0B90 C9	0B56 0B	0B55 62	0B4A 0E	0B49 C7	0B0B 0E	0B0A C7	0AF0 0A	0AEF F6	0A9E 0A	0A9D A1
0A8E 0E	0A8D C7	0A68 0A	0A67 6F	0A63 0E	0A62 C7	0A30 0A	0A2F 36	0A22 0A	0A21 29	0A1B 0A	0A1A 1F	0A10 0A
0A0F 16	0A09 0A	0A08 42	09F8 0E	09F7 C7	09F1 0A	09F0 46	09E7 0A	09E6 59	09DE 0A	09DD 46	09D7 09	09D6 DF
09BD 09	09BC C7	09AF 09	09AE B1	099D 09	099C A9	0995 09	0994 B6	095A 09	0959 61	0948 09	0947 52	093F 09
093A 73	0935 09	0934 3F	092D 09	092C 30	0924 09	0923 27	0921 0E	0920 C7	08F6 08	08F5 FE	08EE 08	08ED F1
08E6 0E	08E5 C7	08C6 08	08C5 C9	0860 08	085F 72	0845 08	0844 4E	083B 08	083A 7F	0795 07	0794 A1	0775 0E
0774 A1	076E 07	076D 70	0714 07	0713 1C	070C 07	070B 0F	0702 0E	0701 A1	06F7 0E	06F6 A1	06E8 0E	06E7 A1
06E2 0E	06E1 A1	06C4 06	06C3 CF	0696 0E	0695 B4	0686 06	0685 90	0676 06	0675 90	05FB 06	05FA 1A	05F7 06
05F6 38	05C8 05	05C7 DD	05C4 05	05C3 D3	05BE 05	05BD E7	05B3 05	05B2 EE	059F 05	059E A2	059B 05	059A A2
0591 05	0590 9C	0573 05	0572 79	056E 05	056D 78	0561 05	0560 87	054F 05	054E 5C	0539 05	0538 4B	052C 05
052B 3A	0525 05	0524 29	0515 05	0514 19	0508 05	0507 5C	04F6 05	04F5 64	04EF 04	04EE F3	04D4 04	04D3 D6
04CB 0E	04CA C0	04C7 04	04C6 CF	04BC 04	04BB BE	0489 04	0488 92	0484 04	0483 8B	046E 04	046D 7B	045C 04
045B 97	0456 04	0455 A3	0450 0E	044F C0	044B 04	044A A3	0442 04	0441 44	043B 04	043A B2	0413 0E	0412 AF
0400 04	03FF 0B	03F8 04	03F7 0B	03F2 03	03F1 F9	03EE 04	03ED 0B	03E7 03	03E6 FB	03E3 04	03E2 14	03CE 04
03CD 0F	03B0 03	03AF FD	03A1 03	03A0 DE	038E 03	038D B7	038A 03	0389 A2	0386 03	0385 8B	0382 03	0381 B1
0377 03	0376 7B	0373 03	0372 D0	0368 03	0367 C3	034C 03	034B 55	0332 03	0331 48	031E 0E	031D AF	0301 03
0300 28	02F5 0E	02F4 AF	02E2 02	02E1 E6	02D3 02	02D2 F6	02BC 02	02BB BE	02AE 02	02AD B2	02A2 02	02A1 F6
029B 02	029A CB	028B 02	028A B6	0284 02	0283 88	0276 02	0275 F6	0270 02	026F CB	026B 02	026A 97	025A 02
0259 5F	0251 02	0250 64	0249 02	0248 64	0240 02	023F 46	01F4 01	01F3 FE	01DB 01	01DA E1	0177 0E	0176 C7
0174 EA	0173 A8	00D4 13	00D3 57	00C9 13	00C8 57	00AE 0E	00AD B4	00A8 0E	00A7 B4	007A 00	0079 96	0075 00
0074 8D	0072 15	0071 58	006E 00	006D 81	0061 00	0060 CB	0059 00	0058 66	0056 00	0055 BD	0052 07	0051 50
0018 15	0017 58	0016 0E	0015 35	0014 0D	0013 F5	0012 00	0011 DA	000F 00	000E D6	000C 0E	000B 75	0009 10
0008 CB	0005 0E	0004 E2	0002 0E	0001 DF								

4.2 GAT13

The ALGOL program GAT13 was originally a default procedure loaded to insure well defined behavior in loop 4. The modification to GAT13 implements the gateway function between loop 4 and loop 5. GAT13 is identical to GAT12 except for differences in loop addresses, logical ID's, etc. and the interface to the gateway being interrupt driven in GAT13.

```
$ MERGE SOURCE/NODE-CNTRL DISK                                                  00000010          0:0000:*
$ -SET SEC +SET GAT -SET HST -SET CRT -SET SDLC -SET TCCF -SET N14              00000100          0:0000:*
$ -SET N14 -SET N15 -SET G12 +SET G13 -SET D13 -SET D17                         00000110          0:0000:*
$  LIST CLIST                                                                   00000200          0:0000:*
BEGIN       %NODAL CONTROL PROGRAM                                              00000300 D        0:0000:*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00000400 D        0:0000:*
PROCEDURE RESTART FORWARD;                                                      00000500 D        0:0000:*
NEW G1, G2;                                                                     00000600 D        0:0000:3
PROCEDURE INTERRUPT FORWARD;                                                    00000700 D        0:0001:1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00000800 D        0:0001:2
DEFINE FETCH = @29#, GTS = @38#, DELTA = @19#, RETURN = @57#,                   00000900 D        0:0001:2
   LSS = <#, LEQ = '#, GEQ = !#, GTR = >#, EQL = =#, NOP = @62#,                00001000 D        0:0001:2
   TRUE = 1#, FALSE = 0#, HEX = HEXDIGITS#, CIO = @26#, ABS = @34#;             00001100 D        0:0001:2
DEFINE NEQ = ^#;                                                                00001200          0:0001:2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00001300 D        0:0001:2
NEW QFLAG := 0, I, ALT , ALTRT := 0, D5, D6 := 18, LQ, SLID, RESENDLIM,         00001400 D        0:0001:2
   QLNGTH, WRTTIME, OLDTIME, NEWTIME, RDA, WTA := 255, FLWCNT := 0,             00001500 D        0:0003:3
   TIMELIM :=10, ACKTIME, SETPRIM := 4, RSTPRIM := 32, SETBKUP := 8,            00001600 D        0:0006:0
   RSTBKUP := 64, ATIMELIM :=  70, WAITFORLB := 0, OFF := 0,QFLAG8:=0,          00001700 D        0:0007:1
   MONITOR := 18, MESSSEQ := -1, PACKSEQ, MASK, Q1, Q2, M, WRT := 0,            00001800 D        0:0009:2
   INFLWCNT := 0, OUTFLWCNT := 0, FULL := 0, SENT := 0, OUTQ;                   00001900 D        0:0011:3
$OMIT = NOT CRT                                                                 00002000 D        0:0013:0
$POP OMIT                                                                       00002300 D        0:0013:0
DEFINE ETX = 4"43"#, CR = 4"4D"#, LEOP = 4"FF"#, LF = 4"4A"#,CLR=4"4C"#,        00002400 D        0:0013:0
   NUL = 4"40"#, FREESIZE = 40#, HOME = 4"54"#, DC1 = 4"51"#,                   00002500 D        0:0013:0
   XOFF = 4"53"#;                                                               00002600 D        0:0013:0
ARRAY INBUF [64], OUTBUF [64], XINBUF [480], XOUTBUF [64],                      00002700 D        0:0013:0
   INQ [8],  XOUTQ [8], XINQ [8] , TEMP [2], PACK [FREESIZE,64],                00002800 D        0:0017:0
   FREE [FREESIZE DIV 4], LIDFAD [64], ACKQ [8], LINE [2];                      00002900 D        0:0020:2
ARRAY ACKSEQ [256];                                                             00003000 D        0:0023:0
$OMIT = NOT GAT                                                                 00003100 D        0:0024:0
ARRAY TASCINT [16], TINTASC [16];                                               00003200 D        0:0024:0
$POP OMIT                                                                       00003300 D        0:0025:0
$OMIT = NOT SEC                                                                 00003400 D        0:0025:0
$POP OMIT                                                                       00003600 D        0:0025:0
$ OMIT = NOT G13                                                                00003610          0:0025:0
ARRAY SET [8];                                                                  00003620          0:0025:0
     PROCEDURE INTASC FORWARD;                                                  00003630          0:0025:2
     PROCEDURE ASCINT FORWARD;                                                  00003640          0:0025:3
$ POP OMIT                                                                      00003650          0:0026:0
POINTER PQ, PF, PI, QP;                                                         00003700 D        0:0026:0
$ OMIT = G13                                                                    00003710          0:0027:0
$ POP OMIT + OMIT = NOT G13                                                     00003810          0:0027:0
DEFINE LIO = @96#, RAM = @97#, STAT = @98#, CIO = @26#, ACK = 1#,               00003820          0:0027:0
$ POP OMIT                                                                      00003830          0:0027:0
   NAK = 128#, WTOKEN = @101#, TIMER = @100#, ENABLE = @35#, BIO =@102#,        00003900 D        0:0027:0
   PIO = @104#, SWITCH = @105#;                                                 00004000 D        0:0027:0
NEW BACKUPSET:=FALSE, PRIMARYSET:=FALSE;                                        00004010 D        0:0027:0
$OMIT = NOT SDLC                                                                00004100 D        0:0027:2
$POP OMIT + OMIT = NOT TCCF                                                     00004400 D        0:0027:2
$POP OMIT                                                                       00004700 D        0:0027:2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00004800 D        0:0027:2
PROCEDURE GETSPACE FORWARD;                                                     00004900 D        0:0027:2
PROCEDURE ENQUEUE FORWARD;                                                      00005000 D        0:0027:3
PROCEDURE ERRORREC ( N); NEW N;                                                 00005100 D        0:0028:0
BEGIN                                                                           00005200 D        0:0029:3
NEW T1;                                                                         00005300 D        0:0029:3
```

```
START OF SEGMENT    2
                    IF N = 4 THEN                                                                    00005400 D        2:0000:0
                    BEGIN                                                                            00005500 D        2:0001:0
                       REPLACE POINTER (PACK [(T1 := GETSPACE), 0]) BY                               00005600 D        2:0002:0
                          (MESSSEQ := (IF MESSSEQ=254 THEN 0 ELSE MESSSEQ + 1)) FOR 1,  %00005700 D        2:0005:0
                          0 FOR 1,     %D2                                                          00005800 D        2:0010:3
                          0 FOR 2,                          %D3,D4                                 00005900 D        2:0011:2
                          MONITOR FOR 1,                        %D5                                00006000 D        2:0012:1
                          SLID FOR 1,                            %D6                               00006100 D        2:0013:1
                          HOME FOR 1,  LF FOR 23,      %HOME , LINEFEEDS                           00006200 D        2:0014:1
                          "NO RESPONSE FROM ",                                                     00006300 D        2:0016:3
                 $OMIT = NOT HST                                                                     00006400 D        2:0018:2
                 $POP OMIT + OMIT = NOT CRT                                                          00006600 D        2:0018:2
                 $POP OMIT + OMIT = NOT SEC                                                          00006800 D        2:0018:2
                 $POP OMIT                                                                           00007000 D        2:0018:2
                          HOME FOR 1, ETX FOR 1;                                                     00007100 D        2:0018:2
                       REPLACE POINTER (PACK [T1, 0]) + 254 BY 59 FOR 1;                             00007200 D        2:0021:1
                       ENQUEUE (ACKQ, T1);                                                           00007300 D        2:0025:1
                    END ELSE RESTART (1);                                                            00007400 D        2:0027:1
                 END;         %ERRORREC                                                              00007500 D        2:0029:2
SEGMENT    2 IS   30 LONG.
                 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00007600 D        0:0031:0
                 PROCEDURE GETSPACE;                                                                 00007700 D        0:0031:0
                 BEGIN                                                                               00007800 D        0:0033:1
                    IF DELTA (POINTER (FREE), PF) LSS 0 THEN RESTART (2);                            00007900 D        0:0033:2
                    GETSPACE := FETCH (PF, 1);                                                       00008000 D        0:0038:0
                    IF DELTA (POINTER (FREE), (PF := PF - 1)) LEQ 5 THEN                             00008100 D        0:0039:2
                       FLWCNT := 1 ELSE FLWCNT := 0;                                                 00008200 D        0:0043:1
                 END;      %GETSPACE                                                                 00008300 D        0:0047:1
                 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00008400 D        0:0047:2
                 PROCEDURE ENQUEUE (A, N); ARRAY A[0]; NEW N;                                        00008500 D        0:0047:2
                 BEGIN                                                                               00008600 D        0:0050:2
                 NEW QHEAD, QTAIL;                                                                   00008700 D        0:0050:0
START OF SEGMENT    3
                    QHEAD := FETCH (POINTER (A), 1);                                                 00008800 D        3:0000:1
                    QTAIL := FETCH (POINTER (A) + 1, 1);                                             00008900 D        3:0002:1
                    IF QTAIL GTR 2 THEN                                                              00009000 D        3:0004:3
                    BEGIN                                                                            00009100 D        3:0005:2
                       REPLACE POINTER (A) + (QTAIL := QTAIL - 1) BY N FOR 1;                        00009200 D        3:0006:2
                       REPLACE POINTER (A) + 1 BY QTAIL FOR 1;                                       00009300 D        3:0010:0
                    END ELSE                                                                         00009400 D        3:0012:2
                    BEGIN                                                                            00009500 D        3:0013:2
                       IF QHEAD LSS QLNGTH - 1 THEN                                                  00009600 D        3:0013:2
                       BEGIN                                                                         00009700 D        3:0014:3
                          FOR I := 0 STEP 1 UNTIL QHEAD - QTAIL DO                                   00009800 D        3:0015:3
                             REPLACE POINTER (A) + (QLNGTH - 1 - I) BY POINTER (A)                   00009900 D        3:0021:0
                                + (QHEAD - I) FOR 1;                                                 00010000 D        3:0023:3
                          REPLACE POINTER (A) BY QLNGTH - 1 FOR 1, QLNGTH - 1 -                      00010100 D        3:0027:2
                             (QHEAD - QTAIL) FOR 1;                                                  00010200 D        3:0030:2
                          ENQUEUE (A, N);                                                            00010300 D        3:0032:1
                       END ELSE                                                                      00010400 D        3:0034:2
                       ERRORREC (4);    %QUEUE FULL                                                 00010500 D        3:0035:2
                    END;                                                                             00010600 D        3:0036:3
                 END;                                                                                00010700 D        3:0036:3
SEGMENT    3 IS   38 LONG.
                 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00010800 D        0:0051:1
                 PROCEDURE DEQUEUE (A, D); ARRAY A [0]; NEW D;                                       00010900 D        0:0051:1
                 BEGIN                                                                               00011000 D        0:0053:2
                 NEW QHEAD;                                                                          00011100 D        0:0053:0
START OF SEGMENT    4
```

```
                  QHEAD := FETCH (POINTER (A), 1);                                                    00011200 D        4:0000:0
                  DEQUEUE := FETCH (POINTER (A) + QHEAD, 1);                                          00011300 D        4:0002:0
                  IF D THEN                                                                           00011400 D        4:0004:3
                     IF QHEAD = 2 THEN REPLACE POINTER (A) BY QLNGTH FOR 1,                           00011500 D        4:0005:1
                        QLNGTH + 1 FOR 1 ELSE                                                         00011600 D        4:0010:0
                     REPLACE POINTER (A) BY QHEAD - 1 FOR 1;                                          00011700 D        4:0012:2
               END;                                                                                    00011800 D        4:0014:3
SEGMENT    4 IS   16 LONG.
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00011900 D        0:0054:1
               PROCEDURE ACKNAK (N); NEW N;                                                           00012000 D        0:0054:1
               BEGIN                                                                                  00012100 D        0:0056:0
               NEW T1, T2, L;                                                                         00012200 I        0:0056:0
START OF SEGMENT    5
               POINTER P;                                                                             00012300 D        5:0000:2
               PROCEDURE FINDPACK (J); NEW J;                                                         00012400 D        5:0000:3
               BEGIN                                                                                  00012500 D        5:0002:3
                  FINDPACK := OUTQ;                                                                   00012600 D        5:0002:3
                  IF J = 1 THEN                                                                       00012700 D        5:0003:3
                  BEGIN                                                                               00012800 D        5:0004:2
                     PACK [OUTQ, 63] := 0;  %MARK PACKET EMPTY                                        00012900 D        5:0005:2
                     FULL := SENT := 0;                                                               00013000 D        5:0008:3
                  END;                                                                                00013100 D        5:0010:2
               END;                                                                                   00013200 D        5:0010:2
               %                                                                                      00013300 D        5:0010:3
               PROCEDURE NOTSENT (M1); NEW M1;                                                        00013400 D        5:0010:3
               BEGIN                                                                                  00013500 D        5:0012:3
                  REPLACE POINTER (PACK [(T1 := GETSPACE), 0]) BY                                     00013600 D        5:0012:3
                     0 FOR 1,        %D1                                                              00013700 D        5:0016:0
                     (MESSSEQ := (IF MESSSEQ=254 THEN 0 ELSE MESSSEQ + 1)) FOR 1,  %D2                00013800 D        5:0016:3
                     0 FOR 2,                          %D3,D4                                         00013900 D        5:0022:2
                     MONITOR FOR 1,                       %D5                                         00014000 D        5:0023:1
                     SLID FOR 1,                           %D6                                        00014100 D        5:0024:1
                     HOME FOR 1,  LF FOR 23,       %HOME , LINEFEEDS                                  00014200 D        5:0025:1
                     "PACKET ",FETCH (P, 1) FOR 3 DIGITS, " OF MESSAGE ",                             00014300 D        5:0027:3
                     FETCH (P + 1, 1) FOR 3 DIGITS, " NOT SENT FROM NODE ", SLID FOR 3                00014400 D        5:0032:3
                     DIGITS, " TO NODE ",  FETCH (P + 4, 1) FOR 3 DIGITS, " ", P                      00014500 D        5:0037:0
                     FOR 18, ETX FOR 1;                                                               00014600          5:0042:0
                  REPLACE POINTER (PACK [T1, 0]) + 254 BY 110 FOR 1;                                  00014700 D        5:0044:3
                  ENQUEUE (ACKQ, T1);                                                                 00014800 D        5:0049:2
                  PACK [OUTQ, 63] := SENT := FULL := WRT := 0;                                        00014900 D        5:0051:3
                  REPLACE (PF := PF + 1) BY OUTQ FOR 1;                                               00015000 D        5:0057:1
               END;   %NOTSENT                                                                       00015100 D        5:0060:2
                  IF N THEN REPLACE (PF := PF + 1) BY FINDPACK (1) FOR 1 ELSE %ACK                    00015200 D        5:0060:3
                  BEGIN                                                                               00015300 D        5:0066:3
                     T1 := FINDPACK (0);                                                              00015400 D        5:0066:3
                     T2 := FETCH ((P := POINTER (PACK [T1, 0])) + 255, 1);                            00015500 D        5:0068:0
                     IF (T2 := T2 + 1) GTR RESENDLIM THEN                                             00015600 D        5:0072:3
                        IF FETCH (P + 2, 1).[5:1] THEN                                                00015700 D        5:0074:2
                           NOTSENT (T1) ELSE                                                          00015800 D        5:0077:2
                        BEGIN                                                                         00015900 D        5:0080:2
                           REPLACE P + 2 BY (FETCH (P + 2, 1) & 1[5:1]) FOR 1; %ALTRT                 00016000 D        5:0080:2
               $OMIT = NOT SEC                                                                        00016100 D        5:0084:1
               $POP OMIT                                                                              00016300 D        5:0084:1
               $OMIT = HST                                                                            00016400 D        5:0084:1
                        REPLACE P + 4 BY ALT FOR 1;                                                   00016500 D        5:0084:1
               $POP OMIT                                                                              00016600 D        5:0086:1
                        REPLACE P + 255 BY 0 FOR 1;  %RESET RESEND COUNT                              00016700 D        5:0086:1
                        SENT := 0;                                                                    00016800 D        5:0088:2
                     END ELSE                                                                         00016900 D        5:0089:2
                  BEGIN                                                                               00017000 D        5:0090:2
```

```
            REPLACE P + 255 BY T2 FOR 1;                                    00017100 D   5:0090:2
            SENT := Ø;                                                      00017200 D   5:0092:3
          END;                                                              00017300 D   5:0093:3
        END;                                                                00017400 D   5:0093:3
      END;                                                                  00017500 D   5:0093:3
SEGMENT    5 IS   95 LONG.
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00017600 D   Ø:0057:1
      PROCEDURE CNTRLMESS (CC); NEW CC;                                     00017700 D   Ø:0057:1
      BEGIN                                                                 00017800 D   Ø:0059:Ø
        IF CC.[9:1] THEN RAM (FETCH (PI + 6, 1) & 1[8:8]) ELSE  %ADD RDNDA D700017900 D   Ø:0059:Ø
        IF CC.[1Ø:1] THEN RAM (FETCH (PI + 6, 1) & 2[8:8]) ELSE  %ADD RDA D7 00018000 D   Ø:0065:Ø
        IF CC.[11:1] THEN RAM (FETCH (PI + 6, 1) & 4[8:8]) ELSE  %ADD WTA D7 00018100 D   Ø:0071:Ø
        IF CC.[12:1] THEN                                                   00018200 D   Ø:0077:Ø
          REPLACE POINTER (LIDFAD) + (FETCH (PI + 6, 1) - 1)% CHANGE FAD AT 00018300 D   Ø:0078:Ø
          BY PI + 7 FOR 1 ELSE %    ... TO D8                               00018400 D   Ø:0081:3
      $OMIT = NOT CRT                                                       00018500 D   Ø:0085:Ø
      $POP OMIT + OMIT = CRT                                                00018700 D   Ø:0085:Ø
        IF CC.[14:1] THEN BEGIN END ELSE                                    00018800 D   Ø:0085:Ø
      $POP OMIT                                                             00018900 D   Ø:0088:Ø
        IF CC.[Ø:1] THEN OUTFLWCNT := NOT (OUTFLWCNT) ELSE                  00019000 D   Ø:0088:Ø
        IF CC.[1:1] THEN RAM (FETCH (PI + 7, 1)) ELSE  %DELETE RDND D8      00019100 D   Ø:0092:2
        IF CC.[2:1] THEN RAM (FETCH (PI + 7, 1)) ELSE  %DELETE RDA D8       00019200 D   Ø:0097:2
        IF CC.[3:1] THEN RAM (FETCH (PI + 7, 1)) ELSE  %DELETE WTA D8       00019300 D   Ø:0102:2
        IF CC.[4:1] THEN SWITCH (SETPRIM) ELSE                              00019400 D   Ø:0107:2
        IF CC.[5:1] THEN                                                    00019500 D   Ø:0111:2
          BEGIN                                                             00019510 D   Ø:0112:2
            SWITCH(RSTPRIM);                                                00019520 D   Ø:0113:2
            PRIMARYSET:=FALSE;                                              00019530 D   Ø:0114:2
          END ELSE                                                          00019540 D   Ø:0115:2
        IF CC.[6:1] THEN SWITCH (SETBKUP) ELSE                              00019600 D   Ø:0116:2
        IF CC.[7:1] THEN                                                    00019700 D   Ø:0120:2
          BEGIN                                                             00019710 D   Ø:0121:2
            SWITCH(RSTBKUP);                                                00019720 D   Ø:0122:2
            BACKUPSET:=FALSE;                                               00019730 D   Ø:0123:2
          END;                                                              00019740 D   Ø:0124:2
      END;                                                                  00019800 D   Ø:0124:2
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00019900 D   Ø:0124:3
      PROCEDURE INPUTQUEUE (L); NEW L;                                      00020000 D   Ø:0124:3
      BEGIN                                                                 00020100 D   Ø:0126:2
      NEW T1, LI;                                                           00020200 D   Ø:0126:2
START OF SEGMENT   6
      PROCEDURE BUILDACK (N); NEW N;                                        00020300 D   6:0000:1
      $OMIT = NOT GAT                                                       00020400 D   6:0002:1
      IF FETCH (PI + 4, 1) = SLID THEN                                      00020500     6:0002:1
      $POP OMIT                                                             00020600 D   6:0004:2
      BEGIN                                                                 00020700 D   6:0004:2
        T1 := GETSPACE;                                                     00020800 D   6:0005:2
        REPLACE POINTER (PACK [T1, Ø]) BY                                   00020900 D   6:0007:Ø
          POINTER (INBUF)  FOR 2,   %D1 PACKET SEQ NO                       00021000 D   6:0009:1
                                    %D2 MESSAGE SEQ NO                      00021100 D   6:0011:Ø
          (IF N = 1 THEN 1 ELSE 128) FOR 1,  %D3 CONTROL ACK OR NAK         00021200 D   6:0011:Ø
          Ø FOR 1,                           % D4 SECOND CONTROL CHAR       00021300 D   6:0015:1
          POINTER (INBUF) + 5 FOR 1,         %D5 DESTINATION LOGICAL ID     00021400 D   6:0016:Ø
          SLID FOR 1;                        %D6 SOURCE LOGICAL ID          00021500 D   6:0018:1
        REPLACE POINTER (PACK [T1, Ø]) + 254 BY 6 FOR 1;                    00021600 D   6:0019:2
        ENQUEUE (ACKQ, T1);                                                 00021700 D   6:0023:3
      END;                                                                  00021800 D   6:0026:Ø
        LI := L.[8:8] - 2;                                                  00021900 D   6:0026:1
        IF L THEN                                                           00022000 D   6:0028:2
        BEGIN                                                               00022100 D   6:0029:Ø
```

```
                $OMIT = GAT                                                                    00022200 D      6:0030:0
                $OMIT = CRT                                                                    00022300 D      6:0030:0
                $POP OMIT                                                                      00022600 D      6:0030:0
                $POP OMIT                                                                      00022700 D      6:0030:0
                        T1 := GETSPACE;                                                        00022800 D      6:0030:0
                        REPLACE POINTER (PACK [T1, 0]) BY POINTER (INBUF) FOR LI;              00022900 D      6:0031:1
                        REPLACE POINTER (PACK [T1, 0]) + 254 BY LI FOR 1;                      00023000 D      6:0035:1
                        ENQUEUE (INQ, T1);                                                     00023100 D      6:0039:1
                $OMIT = GAT                                                                    00023200 D      6:0041:1
                $OMIT = CRT                                                                    00023400 D      6:0041:1
                $POP OMIT                                                                      00023700 D      6:0041:1
                $POP OMIT                                                                      00023800 D      6:0041:1
                   END ELSE BUILDACK (0);    %BUILD NAK MESSAGE                                00023900 D      6:0041:1
                END;                                                                           00024000 D      6:0043:1
SEGMENT    6 IS   44 LONG.
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00024100 D      0:0127:3
                PROCEDURE LOOPINPUT (IB); NEW IB;                                              00024200 D      0:0127:3
                BEGIN                                                                          00024300 D      0:0129:2
                NEW CC, LI;                                                                    00024400 D      0:0129:2
START OF SEGMENT    7
                   PI := POINTER(INBUF);                                                       00024500 D      7:0000:1
                   LI := LIO (IB, INBUF, 0);     %LENGTH OF INPUT BUFFER                      00024600 D      7:0002:1
                   IF (CC := FETCH (PI + 2, 2)) = 0 THEN INPUTQUEUE (LI) ELSE                  00024700 D      7:0004:2
                   BEGIN                                                                       00024800 D      7:0010:0
                $OMIT = NOT GAT                                                                00024900 D      7:0010:0
                      IF (CC.[8:1] OR CC.[15:1]) AND (FETCH (PI + 4, 1) = SLID) THEN           00025000        7:0010:0
                $POP OMIT + OMIT = GAT                                                         00025100 D      7:0014:3
                $POP OMIT                                                                      00025300 D      7:0014:3
                      BEGIN                                                                    00025400 D      7:0014:3
                         ALTRT := CC.[13:1];                                                   00025500 D      7:0015:3
                         IF CC.[8:1] THEN ACKNAK (1) ELSE ACKNAK (0);  %ACK OR NAK             00025600 D      7:0017:2
                      END ELSE                                                                 00025700 D      7:0023:0
                $OMIT = NOT GAT                                                                00025800 D      7:0024:0
                      IF (FETCH (PI, 2) = 4"55AA") AND (FETCH (PI + 4, 1) = SLID) THEN         00025900        7:0024:0
                         CNTRLMESS (CC) ELSE INPUTQUEUE (LI);                                  00026000 D      7:0028:2
                $POP OMIT + OMIT = GAT                                                         00026100 D      7:0033:0
                $POP OMIT                                                                      00026400 D      7:0033:0
                   END;                                                                        00026500 D      7:0033:0
                END;                                                                           00026600 D      7:0033:0
SEGMENT    7 IS   34 LONG.
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00026700 D      0:0130:3
                PROCEDURE LINELOSS (P); POINTER P;                                             00026800 D      0:0130:3
                BEGIN                                                                          00026900 D      0:0132:2
                NEW T1;                                                                        00027000 D      0:0132:2
START OF SEGMENT    8
                   REPLACE POINTER (PACK [(T1 := GETSPACE), 0]) BY                             00027100 D      8:0000:0
                      0 FOR 1,      %D1                                                        00027200 D      8:0003:0
                      (MESSSEQ := (IF MESSSEQ=254 THEN 0 ELSE MESSSEQ + 1)) FOR 1,  %D2        00027300 D      8:0003:3
                      0 FOR 2,                          %D3,D4                               00027400 D      8:0009:2
                      MONITOR FOR 1,                        %D5                              00027500 D      8:0010:1
                      SLID FOR 1,                            %D6                             00027600 D      8:0011:1
                      HOME FOR 1,  LF FOR 23,      %HOME , LINEFEEDS                         00027700 D      8:0012:1
                      "LOSS OF MODULATION ON ",P FOR 7, " LOOP AT NODE ",                      00027800 D      8:0014:3
                      SLID FOR 3 DIGITS, HOME FOR 1, ETX FOR 1;                                00027900 D      8:0019:2
                   REPLACE POINTER (PACK [T1, 0]) + 254 BY 78 FOR 1;                           00028000 D      8:0023:1
                   ENQUEUE (ACKQ, T1);                                                         00028100 D      8:0027:3
                END;    % LINE LOSS                                                           00028200 D      8:0029:3
SEGMENT    8 IS   31 LONG.
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00028300 D      0:0133:3
                PROCEDURE INTERRUPT (INT); NEW INT;                                            00028400 D      0:0133:3
```

```
                BEGIN                                                                00028500 D        0:0136:1
                   BEGIN                                                             00028600 D        0:0136:1
                   NEW STATUS, CHAN;                                                 00028700 D        0:0136:1
START OF SEGMENT    9
                      CHAN := INT.[8:8];                                             00028800 D        9:0000:1
                      STATUS := INT.[0:8];                                           00028900 D        9:0002:0
                      IF CHAN = 8 THEN                                               00029000 D        9:0003:3
                      BEGIN                                                          00029100 D        9:0004:2
                         DO BEGIN                                                    00029200 D        9:0005:2
                            IF STATUS.[0:1] THEN      %IB0F                        00029300 D        9:0005:2
                            BEGIN                                                  00029400 D        9:0006:2
                               STATUS := STATUS & 0 [0:1];                         00029500 D        9:0007:2
                               LOOPINPUT (0);                                      00029600 D        9:0009:1
                            END ELSE                                               00029700 D        9:0010:2
                            IF STATUS.[1:1] THEN      %IB1F                        00029800 D        9:0011:2
                            BEGIN                                                  00029900 D        9:0012:2
                               STATUS := STATUS & 0 [1:1];                         00030000 D        9:0013:2
                               LOOPINPUT  (1);                                     00030100 D        9:0015:1
                            END ELSE                                               00030200 D        9:0016:2
                            IF STATUS.[2:1] THEN      %IB0OVL                      00030300 D        9:0017:2
                            BEGIN                                                  00030400 D        9:0018:2
                               STATUS := STATUS & 0 [2:1];                         00030500 D        9:0019:2
                               LIO (0, INBUF, 0);  %EMPTY BUFFER                   00030600 D        9:0021:1
                            END ELSE                                               00030700 D        9:0023:0
                            IF STATUS.[3:1] THEN      %IB1OVL                      00030800 D        9:0024:0
                            BEGIN                                                  00030900 D        9:0025:0
                               STATUS := STATUS & 0 [3:1];                         00031000 D        9:0026:0
                               LIO (1, INBUF, 0);  %EMPTY BUFFER                   00031100 D        9:0027:3
                            END ELSE                                               00031200 D        9:0029:2
                            IF STATUS.[4:1] THEN      %LNSWPR                      00031300 D        9:0030:2
                            BEGIN                                                  00031400 D        9:0031:2
                               STATUS := STATUS & 0 [4:1];                         00031500 D        9:0032:2
                               IF STATUS.[7:1] AND NOT PRIMARYSET THEN             00031600 D        9:0034:1
                               BEGIN                                               00031700 D        9:0036:1
                                  REPLACE POINTER (LINE) BY "PRIMARY";             00031800 D        9:0037:1
                                  PRIMARYSET:=TRUE;                                00031900 D        9:0040:1
                                  LINELOSS (POINTER (LINE));                       00032000 D        9:0041:1
                               END;                                                00032100 D        9:0043:1
                            END ELSE                                               00032200 D        9:0043:1
                            IF STATUS.[5:1] THEN      %LNSWBK                      00032300 D        9:0044:1
                            BEGIN                                                  00032400 D        9:0045:1
                               STATUS := STATUS & 0 [5:1];                         00032500 D        9:0046:1
                               IF NOT (STATUS.[7:1] OR BACKUPSET) THEN             00032600 D        9:0048:0
                               BEGIN                                               00032700 D        9:0050:0
                                  REPLACE POINTER (LINE) BY "BACK UP";             00032800 D        9:0051:0
                                  LINELOSS (POINTER (LINE));                       00032900 D        9:0054:0
                               BACKUPSET:=TRUE;                                    00032910 D        9:0056:0
                               END;                                                00033000 D        9:0057:0
                            END ELSE                                               00033100 D        9:0057:0
                            IF STATUS.[6:1] THEN   %WTD                            00033200 D        9:0058:0
                            BEGIN                                                  00033300 D        9:0059:0
                               STATUS := STATUS & 0 [6:1];                         00033400 D        9:0060:0
                               WRTTIME := 0; WRT := 0;                             00033500 D        9:0061:3
                            END ELSE                                               00033600 D        9:0063:3
                            IF STATUS.[7:1] THEN    % LINESWITCH                   00033700 D        9:0064:3
                               STATUS := 0;                                        00033800 D        9:0065:3
                         END UNTIL STATUS = 0;                                     00033900 D        9:0067:2
                      END ELSE                                                     00034000 D        9:0069:0
                $OMIT = NOT HST                                                    00034100 D        9:0070:0
                $POP OMIT + OMIT = NOT CRT                                         00040200 D        9:0070:0
```

```
                $POP OMIT + OMIT = NOT GAT                                          00043100 D        9:0070:0
                $ OMIT = G13                                                        00043110          9:0070:0
                $ POP OMIT + OMIT=NOT G13                                           00043202          9:0070:0
                     IF CHAN=2 THEN                                                 00043204          9:0070:0
                     BEGIN                                                          00043206          9:0070:3
                  IF (Q1 := GIO (1, XINBUF, 64)) THEN                               00043208          9:0071:3
                  BEGIN                                                             00043210          9:0074:2
                     REPLACE POINTER(XINBUF) + 256 BY ETX FOR 1;                    00043212          9:0075:2
                     SCAN PI:(PI:=POINTER(XINBUF)+6) WHILE SET;                     00043214          9:0079:0
                     Q1:=DELTA(POINTER(XINBUF),PI)+6;                               00043216          9:0083:2
                     PI := POINTER (XINBUF);                                        00043218          9:0086:2
                *    REPLACE PI + (Q1 - 1) BY ETX FOR 1;                            00043220          9:0088:2
                *    IF (M := FETCH (PI + 2, 2)) = 0 THEN                           00043222          9:0091:2
                     BEGIN                                                          00043224          9:0094:0
                        ENQUEUE (XINQ, (Q2 := GETSPACE));                           00043226          9:0095:0
                        REPLACE POINTER (PACK [Q2, 0]) BY POINTER (XINBUF) FOR Q1;  00043228          9:0098:1
                        REPLACE POINTER (PACK [Q2, 0]) + 254 BY Q1 FOR 1;           00043230          9:0102:3
                     END ELSE                                                       00043232          9:0107:1
                     BEGIN                                                          00043234          9:0108:1
                        IF (M.[8:1] OR M.[15:1]) AND (FETCH (PI + 4, 1) = SLID)  THEN 00043236        9:0108:1
                        BEGIN                                                       00043238          9:0113:2
                           ALTRT := M.[13:1];                                       00043240          9:0114:2
                           IF M.[8:1] THEN ACKNAK (1) ELSE ACKNAK (0);  %ACK OR NAK 00043242         9:0116:2
                        END ELSE                                                    00043244          9:0122:1
                        IF FETCH (PI, 2) = 4"55AA" THEN                             00043246          9:0123:1
                        BEGIN                                                       00043248          9:0125:1
                           INTASC (POINTER (XINBUF) + 6, 2);                        00043250          9:0126:1
                           IF FETCH (PI + 4, 1) = SLID THEN CNTRLMESS (M) ELSE      00043252          9:0129:0
                           BEGIN                                                    00043254          9:0134:3
                              ENQUEUE (XINQ, (Q2 := GETSPACE));                     00043256          9:0134:3
                              REPLACE POINTER (PACK [Q2,0]) BY POINTER (XINBUF) FOR Q1;00043258      9:0138:0
                              REPLACE POINTER (PACK [Q2, 0]) + 254 BY Q1 FOR 1;     00043260          9:0142:2
                           END;                                                     00043262          9:0147:0
                        END ELSE                                                    00043264          9:0147:0
                        IF M.[13:1]  OR M.[8:1] OR M.{                    BEGIN
00043268                    9:0152:1
                           ENQUEUE (XINQ, (Q2 := GETSPACE));                        00043270          9:0153:1
                           REPLACE POINTER (PACK [Q2, 0]) BY POINTER (XINBUF) FOR Q1; 00043272        9:0156:2
                           REPLACE POINTER (PACK [Q2, 0]) + 254 BY Q1 FOR 1;        00043274          9:0161:0
                        END;                                                        00043276          9:0165:2
                     END;                                                           00043278          9:0165:2
                     END;                                                           00043280          9:0165:2
                     END;                                                           00043282          9:0165:2
                $ POP OMIT                                                          00043284          9:0165:2
                $POP OMIT + OMIT = NOT D13                                          00043300 D        9:0165:2
                $POP OMIT + OMIT = NOT D17                                          00043500 D        9:0165:2
                $POP OMIT + OMIT = NOT SEC                                          00043700 D        9:0165:2
                $POP OMIT + OMIT = NOT N14                                          00049900 D        9:0165:2
                $OMIT = NOT SDLC                                                    00050600 D        9:0165:2
                $POP OMIT + OMIT = NOT TCCF                                         00051000 D        9:0165:2
                $POP OMIT                                                           00051400 D        9:0165:2
                $POP OMIT + OMIT = NOT N15                                          00051900 D        9:0165:2
                $OMIT = NOT SDLC                                                    00052600 D        9:0165:2
                $POP OMIT + OMIT = NOT TCCF                                         00053000 D        9:0165:2
                $POP OMIT                                                           00053400 D        9:0165:2
                $POP OMIT                                                           00053900 D        9:0165:2
                   END;                                                             00054000 D        9:0165:2
SEGMENT    9 IS  166 LONG.
                   @58 ;                                                            00054100 D        0:0137:1
                END;                                                                00054200 D        0:0137:3
```

```
                  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00054300 D      0:0138:0
                  PROCEDURE RESTART (X); NEW X;                                          00054400 D      0:0138:0
                  BEGIN                                                                  00054500 D      0:0140:2
                  DEFINE Y = 0#;    %DUMMY DEFINE TO FORCE SEGMENT CREATION              00054600 D      0:0140:2
START OF SEGMENT   10
                    @63 (0) ;  %HALT     LOAD                                            00054700 D     10:0000:*
                  END;       %RESTART                                                    00054800 D     10:0000:2
SEGMENT    10 IS    1 LONG.
                  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00054900 D      0:0141:3
                  PROCEDURE INIT;                                                        00055000 D      0:0141:3
                  BEGIN                                                                  00055100 D      0:0143:1
                  $OMIT = NOT HST                                                        00055200 D      0:0143:2
                  $POP OMIT + OMIT = NOT CRT                                             00055700 D      0:0143:2
                  $POP OMIT + OMIT = NOT GAT                                             00056600 D      0:0143:2
                  $ OMIT = NOT G12                                                       00056610        0:0143:2
                  $POP OMIT + OMIT = NOT G13                                             00056810        0:0143:2
                     SLID := 13;                                                         00056820        0:0143:2
                     RDA := 5;                                                           00056830        0:0144:2
                     REPLACE POINTER (SET) BY 4"FF" FOR 32;                              00056840        0:0145:2
                     SET [ETX.[5:3]] := * & 0 [ETX.[0:5]:1];                             00056850        0:0148:0
                      MASK:=4"0E";                                                       00056860        0:0153:1
                  $ POP OMIT + CMIT = G13                                                00056870        0:0154:1
                  $ POP OMIT                                                             00056910        0:0154:1
                     TASCINT [0] := 4"300F3F0A"; TASCINT [1] := 4"2A3B1C2F";             00057000 D      0:0154:1
                     TASCINT [2] := 4"1D2D2B10"; TASCINT [3] := 4"3A2C1A31";             00057100 D      0:0158:1
                     TASCINT [4] := 4"00010203"; TASCINT [5] := 4"04050607";             00057200 D      0:0162:1
                     TASCINT [6] := 4"08090D2E"; TASCINT [7] := 4"1E3D0E0C";             00057300 D      0:0166:1
                     TASCINT [8] := 4"0B111213"; TASCINT [9] := 4"14151617";             00057400 D      0:0170:1
SEGMENT     1 IS   64 LONG.
                     TASCINT [12] := 4"27282932"; TASCINT [13] := 4"33343536";           00057500 D      0:0174:1
                     TASCINT [10] := 4"18192122"; TASCINT [13] := 4"23242526";           00057600 D      0:0178:1
                     TASCINT [14] := 4"3738391B"; TASCINT [15] := 4"203E3C1F";           00057700 D      0:0182:1
                     TINTASC [0] := 4"30313233"; TINTASC [1] := 4"34353637";             00057800 D      0:0186:1
                     TINTASC [2] := 4"38392340"; TINTASC [3] := 4"3F3A3E21";             00057900 D      0:0190:1
                     TINTASC [4] := 4"2B414243"; TINTASC [5] := 4"44454647";             00058000 D      0:0194:1
                     TINTASC [6] := 4"48492E5B"; TINTASC [7] := 4"26283C5F";             00058100 D      0:0198:1
                     TINTASC [8] := 4"5C4A4B4C"; TINTASC [9] := 4"4D4E4F50";             00058200 D      0:0202:1
                     TINTASC [10] := 4"5152242A"; TINTASC [11] := 4"2D293B27";           00058300 D      0:0206:1
                     TINTASC [12] := 4"202F5354"; TINTASC [13] := 4"55565758";           00058400 D      0:0210:1
                  $ OMIT = NOT G12                                                       00058410        0:0214:1
                  $ POP OMIT + OMIT = NOT G13                                            00058510        0:0214:1
                     GIO (1, INBUF, 0  );                                                00058520        0:0214:1
                  $ POP OMIT                                                             00058530        0:0216:0
                     TINTASC [14] := 4"595A2C25"; TINTASC [15] := 4"5E3D5D22";           00058600 D      0:0216:0
                  $POP OMIT + OMIT = NOT SEC                                             00058700 D      0:0220:0
                  $POP OMIT + OMIT = NOT N14                                             00059900 D      0:0220:0
                  $POP OMIT + OMIT = NOT N15                                             00060300 D      0:0220:0
                  $POP OMIT + OMIT = NOT TCCF                                            00060700 D      0:0220:0
                  $POP OMIT + OMIT = NOT D13                                             00060900 D      0:0220:0
                  $POP OMIT + OMIT = NOT D17                                             00061300 D      0:0220:0
                  $POP OMIT                                                              00061700 D      0:0220:0
                     RESENDLIM := 6 ;          %LIMIT ON RESENDING NAK'ED PACKET         00061800 D      0:0220:0
                  $OMIT = NOT SEC                                                        00061900 D      0:0221:0
                  $POP OMIT + OMIT = SEC                                                 00062100 D      0:0221:0
                     QLNGTH := FREESIZE DIV 4;                                           00062200 D      0:0221:0
                  $POP OMIT                                                              00062300 D      0:0222:0
                     REPLACE POINTER (INQ) BY QLNGTH FOR 1, QLNGTH + 1 FOR 1;%HEAD AND TAI00062400 D      0:0222:0
                     REPLACE POINTER (ACKQ) BY QLNGTH FOR 1, QLNGTH + 1 FOR 1;%HEAD AND T 00062500 D      0:0225:2
                     REPLACE POINTER (XOUTQ) BY QLNGTH FOR 1, QLNGTH + 1 FOR 1;%HEAD AND T00062600 D      0:0229:0
                     REPLACE POINTER (XINQ) BY QLNGTH FOR 1, QLNGTH + 1 FOR 1;%HEAD AND TA00062700 D      0:0232:2
```

```
                  PF := POINTER (FREE);                                                    00062800 D        0:0236:0
                  FOR G1 := 0 STEP 1 UNTIL FREESIZE - 1 DO PACK [G1, 63] := 0;             00062900 D        0:0238:0
                  FOR G1 := 0 STEP 1 UNTIL FREESIZE - 2 DO REPLACE PF:PF BY G1 FOR 1;      00063000 D        0:0246:2
                  FOR G1 := 0 STEP 1 UNTIL 255 DO ACKSEQ [G1] := 255;                      00063100 D        0:0254:1
                  PF := PF - 1;                                                            00063200 D        0:0262:1
                  REPLACE POINTER (LIDFAD) BY 3 FOR 12,  %LID/FAD TABLE                   00063300 D        0:0264:1
                     5 FOR 1,  1 FOR 1, 2 FOR 1, 4 FOR 1, 6 FOR 1, 7 FOR 1, 8 FOR 1,       00063400 D        0:0266:0
                     5 FOR 9, 0 FOR 12, 1 FOR 5, 0 FOR 15, 2 FOR 5, 0 FOR 15,            00063500 D        0:0271:1
                     4 FOR 5, 0 FOR 15, 5 FOR 5, 0 FOR 150;                              00063600 D        0:0275:3
                  FOR G1 := 0 STEP 1 UNTIL 255 DO RAM (G1 & 8[8:8]);                      00063700 D        0:0279:2
                  RAM (RDA & 2[8:8]);    %INITALIZE ADDRESS COMPARISON RAM                 00063800 D        0:0287:1
                  RAM (WTA & 4[8:8]);                                                      00063900 D        0:0289:1
                  ENABLE (MASK);                                                           00064000 D        0:0291:1
                  NEWTIME := TIMER (0);                                                    00064100 D        0:0292:1
               END;                                                                        00064200 D        0:0293:2
               $OMIT = NOT GAT                                                             00064300 D        0:0293:3
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00064400 D        0:0293:3
               PROCEDURE ASCINT (PC, N); POINTER PC; NEW N;                                00064500 D        0:0293:3
               BEGIN                                                                       00064600 D        0:0296:2
               NEW CHAR, OFFSET;                                                           00064700 D        0:0296:1
START OF SEGMENT   12
                  FOR OFFSET := 0 STEP 1 UNTIL N - 1 DO                                    00064800 D       12:0000:1
                  BEGIN                                                                    00064900 D       12:0004:3
                     CHAR := FETCH (PC + OFFSET, 1);                                       00065000 D       12:0004:3
                     IF CHAR LSS 4"20" THEN REPLACE PC + OFFSET BY CHAR + 4"40" FOR 1      00065100 D       12:0006:3
                     ELSE IF CHAR LSS 4"60" THEN                                           00065200 D       12:0010:3
                        REPLACE PC + OFFSET BY POINTER(TASCINT) + (CHAR - 4"20") FOR 1;    00065300 D       12:0013:3
                  END;                                                                     00065400 D       12:0018:3
               END;     %ASCINT                                                          00065500 D       12:0019:3
SEGMENT   12 IS   21 LONG.
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00065600 D        0:0297:2
               PROCEDURE INTASC (PC, N); POINTER PC; NEW N;                                00065700 D        0:0297:2
               BEGIN                                                                       00065800 D        0:0300:1
               NEW CHAR, OFFSET;                                                           00065900 D        0:0300:0
START OF SEGMENT   13
                  FOR OFFSET := 0 STEP 1 UNTIL N - 1 DO                                    00066000 D       13:0000:1
                  BEGIN                                                                    00066100 D       13:0004:3
                     CHAR := FETCH (PC + OFFSET, 1);                                       00066200 D       13:0004:3
                     IF CHAR LSS 4"40" THEN                                                00066300 D       13:0006:3
                        REPLACE PC + OFFSET BY POINTER (TINTASC) + CHAR FOR 1 ELSE         00066400 D       13:0008:0
                     IF CHAR LSS 4"60" THEN REPLACE PC + OFFSET BY CHAR - 4"40" FOR 1;     00066500 D       13:0013:2
                  END;                                                                     00066600 D       13:0018:3
               END;       %INTASC                                                         00066700 D       13:0019:3
SEGMENT   13 IS   21 LONG.
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%00066800 D        0:0301:1
               $POP OMIT                                                                   00066900 D        0:0301:1
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  00067000 D        0:0301:1
               INIT;                                                                       00067100 D        0:0301:1
               WHILE TRUE DO                                                               00067200 D        0:0302:1
               BEGIN                                                                       00067300 D        0:0302:2
                  ENABLE (4);                                                              00067400 D        0:0303:2
                  IF FETCH (POINTER (INQ), 1) GEQ FETCH (POINTER (INQ) + 1, 1) THEN        00067500 D        0:0304:1
                  BEGIN                                                                    00067600 D        0:0308:0
                     ENQUEUE (XOUTQ, (Q2 := DEQUEUE (INQ, 1)));                            00067700 D        0:0309:0
               $OMIT = NOT HST                                                             00067800 D        0:0313:1
               $POP OMIT + OMIT = NOT CRT                                                  00068500 D        0:0313:1
               $POP OMIT + OMIT = NOT SEC                                                  00069100 D        0:0313:1
               $POP OMIT + OMIT = NOT N14                                                  00070100 D        0:0313:1
               $POP OMIT + OMIT = NOT N15                                                  00070300 D        0:0313:1
               $POP OMIT                                                                   00070500 D        0:0313:1
```

```
      ENABLE (MASK);                                                                00070600 D        0:0313:1
   END ELSE ENABLE (MASK);                                                          00070700 D        0:0314:1
   IF NOT(STAT (8).[2:1]) THEN                                                      00070800 D        0:0316:1
   BEGIN                                                                            00070900 D        0:0317:3
      ENABLE (4);                                                                   00071000 D        0:0318:3
      IF FETCH (POINTER (ACKQ), 1) GEQ FETCH (POINTER (ACKQ) + 1,1) THEN            00071100 D        0:0319:2
      BEGIN                                                                         00071200 D        0:0323:1
         LQ := DEQUEUE (ACKQ, 1);                                                   00071300 D        0:0324:1
         REPLACE POINTER (OUTBUF) BY POINTER (PACK [LQ, 0]) FOR                     00071400 D        0:0326:3
         (Q1 := FETCH (POINTER (PACK [LQ, 0]) + 254, 1)),                           00071500 D        0:0330:0
            0 FOR 1,   %BDLC CNTRL CHAR                                             00071600 D        0:0335:0
            POINTER (LIDFAD) + (FETCH (POINTER (OUTBUF) + 4,1)-1) FOR 1;            00071700 D        0:0335:3
         REPLACE (PF := PF + 1) BY LQ FOR 1;                                        00071800 D        0:0340:1
         PACK [LQ, 63] := 0;    %MARK PACKET EMPTY                                  00071900 D        0:0343:2
         LIO (0, OUTBUF, -(Q1 + 2));                                                00072000 D        0:0346:3
         WRTTIME := 0;                                                              00072100 D        0:0349:2
         WRT := 1;                                                                  00072200 D        0:0350:2
         ENABLE (MASK);                                                             00072300 D        0:0351:2
      END ELSE                                                                      00072400 D        0:0352:2
      IF NOT (OUTFLWCNT) THEN                                                       00072500 D        0:0353:2
      IF FULL AND NOT SENT THEN                                                     00072600 D        0:0354:1
      BEGIN                                                                         00072700 D        0:0356:3
         ENABLE(MASK);                                                              00072800 D        0:0357:3
         REPLACE POINTER (OUTBUF) BY POINTER (PACK [OUTQ, 0]) FOR                   00072900 D        0:0358:3
            (Q2 := FETCH (POINTER (PACK [OUTQ, 0]) + 254, 1)),                      00073000 D        0:0362:0
            0 FOR 1,     % BDLC CNTRL CHAR                                          00073100 D        0:0367:0
            POINTER (LIDFAD) + (FETCH (POINTER (OUTBUF) + 4, 1) - 1)                00073200 D        0:0367:3
               FOR 1;                                                               00073300 D        0:0371:0
         SENT := 1;                                                                 00073400 D        0:0372:1
$OMIT = NOT GAT                                                                     00073500 D        0:0373:1
         SENT := FULL := 0;                                                         00073600 D        0:0373:1
         ENABLE (4);                                                                00073700 D        0:0375:0
         REPLACE (PF := PF + 1) BY OUTQ FOR 1;                                      00073800 D        0:0375:3
$VOID                                                                               00073900          0:0379:0
         PACK [OUTQ, 63] := 0;                                                      00074000 D        0:0379:0
$POP OMIT                                                                           00074100 D        0:0382:1
         LIO (0, OUTBUF, -(Q2 + 2));                                                00074200 D        0:0382:1
         ACKTIME := 0;                                                              00074300 D        0:0385:0
         WRTTIME := 0;                                                              00074400 D        0:0386:0
         WRT := 1;                                                                  00074500 D        0:0387:0
         ENABLE (MASK);                                                             00074600 D        0:0388:0
      END ELSE ENABLE(MASK);                                                        00074700 D        0:0389:0
   END;                                                                             00074800 D        0:0391:0
   ENABLE (4);                                                                      00074900 D        0:0391:0
   IF FETCH (POINTER (XINQ), 1) GEQ FETCH (POINTER (XINQ) + 1,1) AND                00075000 D        0:0391:3
      NOT FULL THEN                                                                 00075100 D        0:0395:2
   BEGIN                                                                            00075200 D        0:0396:2
      OUTQ := DEQUEUE (XINQ,1);                                                     00075300 D        0:0397:2
      FULL := 1;                                                                    00075400 D        0:0400:0
      ENABLE (MASK);                                                                00075500 D        0:0401:0
   END ELSE ENABLE (MASK);                                                          00075600 D        0:0402:0
$OMIT = NOT CRT                                                                     00075700 D        0:0404:0
$POP OMIT + OMIT = NOT GAT                                                          00077300 D        0:0404:0
$ OMIT = G13                                                                        00077310          0:0404:0
$ POP OMIT                                                                          00081010          0:0404:0
   IF FETCH (POINTER (XOUTQ), 1) GEQ FETCH(POINTER(XOUTQ) + 1, 1) THEN              00081100 D        0:0404:0
   BEGIN                                                                            00081200 D        0:0407:3
      Q1 := DEQUEUE (XOUTQ, 0);                                                     00081300 D        0:0408:3
      REPLACE POINTER (XOUTBUF) BY POINTER (PACK [Q1, 0]) FOR                       00081400 D        0:0411:1
         FETCH (POINTER (PACK [Q1, 0]) + 254, 1), LEOP FOR 1;                       00081500 D        0:0414:2
```

```
                    IF XOUTBUF [0].[16:16] = 4"55AA" THEN ASCINT( POINTER (XOUTBUF) +
                       6, 2);
                 $ OMIT = G13
                 $ POP OMIT + OMIT = NOT G13
                    IF GIO (1, XOUTBUF, -(FETCH (POINTER (PACK [Q1, 0]) + 254, 1)
                       + 4) DIV 4 ) THEN
                 $ POP OMIT
                    BEGIN
                       ENABLE (4);
                       DEQUEUE (XOUTQ, 1);
                       REPLACE (PF := PF + 1) BY Q1 FOR 1;
                       ENABLE (MASK);
                    END;
                 END;
                 $POP OMIT + OMIT = NOT SEC
                 $POP OMIT + OMIT = NOT N14
                 $POP OMIT + OMIT = NOT N15
                 $POP OMIT + OMIT = NOT CRT
                 $POP OMIT + OMIT = NOT HST
                 $POP OMIT
                 OLDTIME := NEWTIME;
                 NEWTIME := TIMER (0);
                 WRTTIME := WRTTIME + (NEWTIME - OLDTIME);
                 IF WRTTIME GTR TIMELIM AND WRT THEN
                 BEGIN
                    WTOKEN (0);   %IF NO WRITE TOKEN WITHIN TIME LIMIT,ISSUE ONE
                    WRTTIME := 0;
                 END;
                 $OMIT = GAT
                 $POP OMIT
                 ENABLE (MASK);
                 $OMIT = NOT SEC
                 $POP OMIT
                 END;
                 END ?
SEGMENT   11 IS   28 LONG.
SEGMENT    0 IS  454 LONG.
 0 ERRORS    609 CARDS.   4142 TOKENS.      7507 RULES.       158 SECONDS.
 PROGRAM SIZE =   1043 WORDS   26 DISK SECTORS.
```

```
00081600 D        0:0420:1
00081700 D        0:0425:1
00081710          0:0426:2
00081810          0:0426:2
00081820          0:0426:2
00081830          0:0431:0
00081840          0:0432:3
00081900 D        0:0432:3
00082000 D        0:0433:3
00082100 D        0:0434:2
00082200 D        0:0436:2
00082300 D        0:0439:3
00082400 D        0:0440:3
00082500 D        0:0440:3
00082600 D        0:0440:3
00086000 D        0:0440:3
00087000 D        0:0440:3
00088000 D        0:0440:3
00089700 D        0:0440:3
00090000 D        0:0440:3
00090100 D        0:0440:3
00090200 D        0:0442:0
00090300 D        0:0443:1
00090400 D        0:0446:0
00090500 D        0:0448:0
00090600 D        0:0449:0
00090700          0:0449:3
00090800 D        0:0450:3
00090900 D        0:0450:3
00092000 D        0:0450:3
00092100 D        0:0450:3
00092200 D        0:0451:3
00093700 D        0:0451:3
00093800 D        0:0451:3
00093900 D        0:0452:3
```

4.3 ESMD User Language

Additives were made to the User Language to send forms control characters to node 25 which is listed as a CRT but is in fact a DECWriter. A modification was also made to increment the packet numbers in USRLNG messages.

These changes have no effect on the operation of loop 4.

```
                $MERGE USER/P0000 DISK                                                                  00000010          0:0000:*
                %                                                                                       00000100  D       0:0000:*
                %     USER LANGUAGE FOR B776 IN ALGOL                                                   00000200  D       0:0000:*
                %                                                                                       00000300  D       0:0000:*
                                                                                                        00000400  D       0:0000:*
                %     FILE USER/P0000                                                                   00000500  D       0:0000:*
                % OBJECT FILE USRLNG                                                                    00000600  D       0:0000:*
                BEGIN                                                                                   00000700  D       0:0000:*
                $LIST                                                                                   00000800          0:0000:*
                    FILE CARD(CARDS);                                                                   00000900  D       0:0000:*
START OF SEGMENT     1
                    FILE CRTX(CRT,POINTER(","),300);                                                    00001000  D       1:0004:1
                    FILE CKTD(DISK,POINTER("USER/CKTD."),20,180);                                       00001100  D       1:0010:2
                    FILE TNKD(DISK,POINTER("USER/TNKD."),20,180);                                       00001200  D       1:0017:0
                    FILE CDIR(DISK,POINTER("USER/CDIR."),20,180);                                       00001300  D       1:0023:2
                    FILE TDIR(DISK,POINTER("USER/TDIR."),20,180);                                       00001400  D       1:0030:0
                    FILE INFO(DISK,POINTER("USER/INFO."),20,180);  %396                                 00001500  D       1:0036:2
                    FILE MSG(DISK,POINTER("USRLN/MSG."),20,180);  %211                                  00001600  D       1:0043:0
                BEGIN                                                                                   00001700  D       1:0049:2
                    ARRAY ST[4,9],MSGNO[4],DEVST[6],OFIL[20],SET[8],DI[100],                            00001800  D       1:0049:2
START OF SEGMENT     3
                          ICODE[80],MOUT[21],OUTBUF[64],ICON[80],IFAC[3],ICX[2];                        00001900  D       3:0003:1
                    POINTER P1,PIC,PMT,PCN,P3,PICM;                                                     00002000  D       3:0007:3
                    NEW J,I,NRCNO,NOREC,NWRTS,NN,N1,N2,K,ICFLG,PS;                                      00002100  D       3:0009:1
                    NEW IND,FORMF,BADH;                                                                 00002200  D       3:0012:0
                    NEW SEQ := -1 ;                                                                     00002210          3:0012:3
                                                                                                        00002300  D       3:0013:1
                    DEFINE NUL= 4"40" FOR 1 #,                                                          00002400  D       3:0013:1
                           NULX(X)=4"40" FOR X #,                                                       00002500  D       3:0013:1
                            SEQNO = 0 FOR 1 ,(SEQ := (SEQ + 1) MOD 126) FOR 1#,                         00002510          3:0013:1
                            ETX= 4"43" FOR 1 #,                                                         00002600  D       3:0013:1
                            LF = 4"4A" FOR 1 #,                                                         00002700  D       3:0013:1
                            LFX(X)= HOME, 4"4A" FOR X #,                                                00002800  D       3:0013:1
                            CLEAR= 4"4C" FOR 1, 4"40" FOR 2 #,                                          00002900  D       3:0013:1
                            CR = 4"4D" FOR 1,4"40" FOR 2 #,                                             00003000  D       3:0013:1
                            STAYREC=4"51" FOR 1 #,                                                      00003100  D       3:0013:1
                            LE = 4"59" FOR 1 #,                                                         00003200  D       3:0013:1
                            HOME = 4"54" FOR 1 #,                                                       00003300  D       3:0013:1
                            RLF= 4"53" FOR 1, 4"40" FOR 2 #,                                            00003400  D       3:0013:1
                            US = 4"5F" FOR 1 #,                                                         00003500  D       3:0013:1
                            RS = 4"5E" FOR 1 #,                                                         00003600  D       3:0013:1
                            SETFORMS = 4"52" FOR 1, 4"40" FOR 1 #,                                      00003700  D       3:0013:1
                            BS = 4"48" FOR 1, 4"40" FOR 2 #,                                            00003800  D       3:0013:1
                            ETXAT(P)= (FETCH(P,1)= 4"43") #,                                            00003900  D       3:0013:1
                           IC4W=POINTER(ICX) FOR 2#,                                                    00004000  D       3:0013:1
                           IMC4=4"04" FOR 1,4"10" FOR 1#,                                               00004100  D       3:0013:1
                           IC8W=POINTER(ICX)+2 FOR 2#,                                                  00004200  D       3:0013:1
                           IMC8=4"08" FOR 1,4"10" FOR 1#,                                               00004300  D       3:0013:1
                           IC10W=POINTER(ICX)+4 FOR 2#,                                                 00004400  D       3:0013:1
                           IMC18=4"12" FOR 1,4"10" FOR 1#,                                              00004500  D       3:0013:1
                           IC25W=POINTER(ICX)+6 FOR 2#,                                                 00004600  D       3:0013:1
                           IMC25=4"19" FOR 1, 4"10" FOR 1#,                                             00004700  D       3:0013:1
                           ICVA=4"55" FOR 1,4"AA" FOR 1#,                                               00004800  D       3:0013:1
                           LEOP=4"FF" FOR 2#,                                                           00004900  D       3:0013:1
                           ZER=4"00" FOR 2#,                                                            00005000  D       3:0013:1
                           HEAD1="LIO   2   3   4   5   6   7   8   9  10"#,                            00005100  D       3:0013:1
                           HEAD2="  11  12  13  14  15  16  17  18  19  20"#,                            00005200  D       3:0013:1
                            TR(X)=                                                                      00005300  D       3:0013:1
                              BEGIN                                                                     00005400  D       3:0013:2
                $ OMIT                                                                                  00005500  D       3:0013:2
                $ POP OMIT                                                                              00005900  D       3:0013:2
                              END # ;                                                                   00006000  D       3:0013:2
                                                                                                        00006100  D       3:0013:1
                    PROCEDURE P1000;                                                                    00006200  D       3:0013:1
                %                                                                                       00006300  D       3:0015:0
                %     CRT TO CRT MODE OF OPERATION                                                      00006400  D       3:0015:0
                %                                                                                       00006500  D       3:0015:0
                    BEGIN                                                                               00006600  D       3:0015:0
                    POINTER PX;                                                                         00006700  D       3:0015:1
START OF SEGMENT     4
                    IF ST[IND,1] = 1000 THEN                                                            00006800  D       4:0000:0
                      BEGIN                                                                             00006900  D       4:0002:3
                        ST[IND,2] := INT(PIC,2);                                                        00007000  D       4:0003:3
                        IF ST[IND,2]=4 OR ST[IND,2]=8 OR ST[IND,2]=18                                   00007100  D       4:0007:1
                          OR ST[IND,2]=25 THEN                                                          00007200  D       4:0014:2
                            BEGIN                                                                       00007300  D       4:0018:0
                            NRCNO:= 14;                                                                 00007400  D       4:0019:0
```

```
                    NOREC:= 1;
                    ST[IND,1]:= 1003;
                    END
                  ELSE BEGIN
                    NRCNO:= 11;
                    NOREC:= 3;
                    ST[IND,1]:= 1000;
                    END;
                END
                ELSE IF ST[IND,1]= 1004 THEN
                  BEGIN
                  I:= INT(PIC,1);
                  IF I=1 THEN
                    BEGIN
                    NRCNO:=14;
                    NOREC:= 1;
                    ST[IND,1]:= 1003;
                    END
                  ELSE IF I=2 THEN
                    BEGIN
                    NRCNO:= 12;
                    NOREC:= 2;
                    ST[IND,1]:= 1000;
                    END
                  ELSE IF I=3 THEN
                    BEGIN
                    NRCNO:= 21;
                    NOREC:= 1;
                    ST[IND,1]:= 9999;
                    END
                  ELSE IF I= 4 THEN
                    BEGIN
                    NRCNO:= 5;
                    NOREC:= 5;
                    ST[IND,1]:= 0004;
                    END
                  ELSE BEGIN
                    NRCNO:= 11;
                    NOREC:= 1;
                    END;
                  END
                ELSE BEGIN    % CONNECTOR W(P,1-108)
                    REPLACE PCN BY LEOP,SEQNO,ZER;
                  REPLACE PX:PCN+8 BY LFX[16],PIC WHILE SET,HOME,ETX;
                  TR(DELTA(PCN,PX));
                  IF ST[IND,2]=4 THEN
                        BEGIN
                        REPLACE PCN+6 BY INC4;
                        REPLACE PMT BY "MSG TO CRT  ND= 4",NULX(63);
                        END
                  ELSE IF ST[IND,2]=18 THEN
                        BEGIN
                        REPLACE PCN+6 BY INC18;
                        REPLACE PMT BY "MSG TO CRT  ND= 18",NULX(62);
                        END
                  ELSE IF ST[IND,2]=25 THEN
                        BEGIN
                REPLACE PX:PCN+8 BY CR,LF,PIC WHILE SET,CR,LF,ETX;
                        REPLACE PCN+6 BY INC25;
                        REPLACE PMT BY "MSG TO CRT  ND=25",NULX(62);
                        END
                        ELSE BEGIN
                        REPLACE PCN+6 BY INC8;
                        REPLACE PMT BY "MSG TO CRT ND=8",NULX(63);
                        END;
                  WRITE  (MSG[33],MOUT);
                  FOR J:= 1 STEP 1 UNTIL 5 DO
                        BEGIN
                        READ(MSG[J+13],MOUT);
                        WRITE(MSG[J+33],MOUT);
                        END;
                  NRCNO:= 34;
                  NOREC:= 6;
                  ICFLG:= 1;
                  ST[IND,1] := 1004;
                  END;
                END;
SEGMENT    4 IS  160 LONG.
                %
                %         SYSTEM INQUIRY MODE OF OPERATION -MODE 2
```

```
00007500 D      4:0020:0
00007600 D      4:0021:0
00007700 D      4:0024:0
00007800 D      4:0024:0
00007900 D      4:0025:0
00008000 D      4:0026:0
00008100 D      4:0027:0
00008200 D      4:0030:0
00008300 D      4:0030:0
00008400 D      4:0030:0
00008500 D      4:0033:3
00008600 D      4:0034:3
00008700 D      4:0036:2
00008800 D      4:0037:2
00008900 D      4:0038:2
00009000 D      4:0039:2
00009100 D      4:0040:2
00009200 D      4:0041:2
00009300 D      4:0042:2
00009400 D      4:0045:2
00009500 D      4:0046:2
00009600 D      4:0047:2
00009700 D      4:0048:2
00009800 D      4:0051:2
00009900 D      4:0051:2
00010000 D      4:0053:2
00010100 D      4:0054:2
00010200 D      4:0055:2
00010300 D      4:0056:2
00010400 D      4:0059:2
00010500 D      4:0059:2
00010600 D      4:0061:2
00010700 D      4:0062:2
00010800 D      4:0063:2
00010900 D      4:0064:2
00011000 D      4:0067:0
00011100 D      4:0067:0
00011200 D      4:0068:0
00011300 D      4:0069:0
00011400 D      4:0070:0
00011500 D      4:0070:0
00011600 D      4:0070:0
00011700        4:0071:0
00011800 D      4:0077:2
00011900 D      4:0086:0
00012000 D      4:0086:0
00012100 D      4:0088:2
00012200 D      4:0089:2
00012300 D      4:0092:1
00012400 D      4:0096:0
00012500 D      4:0096:0
00012600 D      4:0099:2
00012700 D      4:0100:2
00012800 D      4:0103:1
00012900 D      4:0107:0
00013000 D      4:0107:0
00013100 D      4:0110:2
00013150        4:0111:2
00013200 D      4:0123:3
00013300 D      4:0126:2
00013400 D      4:0130:1
00013500 D      4:0130:1
00013600 D      4:0131:1
00013700 D      4:0134:0
00013800 D      4:0137:3
00013900 D      4:0137:3
00014000 D      4:0140:3
00014100 D      4:0145:2
00014200 D      4:0145:2
00014300 D      4:0149:1
00014400 D      4:0153:0
00014500 D      4:0153:2
00014600 D      4:0154:2
00014700 D      4:0155:2
00014800 D      4:0156:2
00014900 D      4:0159:2
00015000 D      4:0159:2

00015100 D      3:0016:2
00015200 D      3:0016:2
```

```
                %
                PROCEDURE P2000;                                                    00015300 D      3:0016:2
                  BEGIN COMMENT SYSTEM INQUIRY MODE OF OPERATION;                   00015400 D      3:0016:2
                  NEW ND,NR,JJ,NNR,MNR;                                             00015500 D      3:0018:1
START OF SEGMENT     5                                                            00015600 D      3:0018:2
                   ARRAY OFIL[20];                                                  00015700 D      5:0001:0
                  POINTER PX,POF;                                                   00015800 D      5:0001:2
                   POF:=POINTER(OFIL);                                              00015900 D      5:0002:0
                  IF ST[IND,1] = 2000 THEN                                          00016000 D      5:0003:2
                    BEGIN                                                           00016100 D      5:0006:1
                   TR(1);                                                           00016200 D      5:0007:1
                    I := INT(PIC,1);                                                00016300 D      5:0007:1
                    IF I LEQ 0 OR I GEQ 5 THEN                                      00016400 D      5:0009:0
                      BEGIN                                                         00016500 D      5:0011:1
                      NRCNO := 11;                                                  00016600 D      5:0012:1
                      NOREC := 1;                                                   00016700 D      5:0013:1
                      END                                                           00016800 D      5:0014:1
                    ELSE IF I=1 THEN                                                00016900 D      5:0014:1
                      BEGIN                                                         00017000 D      5:0016:1
                      FOR J:= 0 STEP 1 UNTIL 11 DO                                  00017100 D      5:0017:1
                        BEGIN                                                       00017200 D      5:0022:0
                        READ(INFO[J],MOUT);                                         00017300 D      5:0022:0
                        WRITE(MSG[J+33],MOUT);                                      00017400 D      5:0025:1
                        END;                                                        00017500 D      5:0029:0
                      NRCNO:= 34;                                                   00017600 D      5:0029:2
                      NOREC := 12;                                                  00017700 D      5:0030:2
                      ST[IND,1] := 2004;                                            00017800 D      5:0031:2
                      END                                                           00017900 D      5:0034:2
                    ELSE BEGIN                  % I NE 1                          00018000 D      5:0034:2
                      ST[IND,3] :=I;                                                00018100 D      5:0035:2
                      NRCNO := 27;                                                  00018200 D      5:0038:2
                      NOREC := 3;                                                   00018300 D      5:0039:2
                      ST[IND,1] := 2003;                                            00018400 D      5:0040:2
                      END;                                                          00018500 D      5:0043:2
                    END                                                             00018600 D      5:0043:2
                    ELSE IF ST[IND,1] = 2004 THEN                                   00018700 D      5:0043:2
                      BEGIN                                                         00018800 D      5:0047:1
                   TR(2);                                                           00018900 D      5:0048:1
                      NRCNO:=30;                                                    00019000 D      5:0048:1
                      NOREC:= 4;                                                    00019100 D      5:0049:1
                      ST[IND,1]:= 2005;                                             00019200 D      5:0050:1
                      END                                                           00019300 D      5:0053:1
                    ELSE IF ST[IND,1]=2005 THEN                                     00019400 D      5:0053:1
                      BEGIN                                                         00019500 D      5:0057:0
                   TR(3);                                                           00019600 D      5:0058:0
                      I:= INT(PIC,1);                                               00019700 D      5:0058:0
                      IF I=1 THEN                                                   00019800 D      5:0059:3
                        BEGIN                                                       00019900 D      5:0060:3
                        NRCNO:= 22;                                                 00020000 D      5:0061:3
                        NOREC:= 5;                                                  00020100 D      5:0062:3
                        ST[IND,1]:=2000;                                            00020200 D      5:0063:3
                        END                                                         00020300 D      5:0066:3
                      ELSE IF I=2 THEN                                              00020400 D      5:0066:3
                        BEGIN                                                       00020500 D      5:0068:3
                        NRCNO := 21;                                                00020600 D      5:0069:3
                        NOREC := 1;                                                 00020700 D      5:0070:3
                        ST[IND,1] := 9999;                                          00020800 D      5:0071:3
                        END                                                         00020900 D      5:0074:3
                      ELSE IF I=3 THEN                                              00021000 D      5:0074:3
                        BEGIN                                                       00021100 D      5:0076:3
                        NRCNO := 5;                                                 00021200 D      5:0077:3
                        NOREC:= 5;                                                  00021300 D      5:0078:3
                        ST[IND,1]:= 0004;                                           00021400 D      5:0079:3
                        END                                                         00021500 D      5:0082:1
                      ELSE BEGIN                % I NE 1,2,OR 3                   00021600 D      5:0082:1
                        NRCNO := 11;                                                00021700 D      5:0083:1
                        NOREC := 1;                                                 00021800 D      5:0084:1
                        END;                                                        00021900 D      5:0085:1
                      END                                                           00022000 D      5:0085:1
                    ELSE BEGIN                  % ST[1,1] NE 2000,2004,2005       00022100 D      5:0085:1
                   TR(4);                                                           00022200 D      5:0086:1
                      IF PIC = "NDI" THEN                                           00022300 D      5:0086:1
                        BEGIN                                                       00022400 D      5:0088:1
                        FOR J:=0 STEP 1 UNTIL 11 DO                                 00022500 D      5:0089:1
                          BEGIN                                                     00022600 D      5:0094:0
                          READ(INFO[J],MOUT);                                       00022700 D      5:0094:0
                          WRITE (MSG[J+33],MOUT);                                   00022800 D      5:0097:1
                          END;                                                      00022900 D      5:0101:0
                        NRCNO := 34;                                                00023000 D      5:0101:2
                        NOREC:= 12;                                                 00023100 D      5:0102:2
```

```
                    ST[IND,1]:= 2004;                                                    00023200 D        5:0103:2
                    END                                                                  00023300 D        5:0106:2
                                                                                         00023400 D        5:0106:2
                ELSE BEGIN              % ND HOLDS A NUMERIC VALUE                       00023500 D        5:0106:2
               TR(5);                                                                    00023600 D        5:0107:2
                  SCAN PX:PIC WHILE "0123456789";                                        00023700 D        5:0107:2
                  ND:= JJ:= INT(PIC,DELTA(PIC,PX));                                      00023800 D        5:0110:0
                  IF JJ=0 OR JJ GEQ 29 THEN                                              00023900 D        5:0112:3
                    BEGIN                                                                00024000 D        5:0114:1
                    NRCNO:= 1;                                                           00024100 D        5:0115:1
                    NOREC := 1;                                                          00024200 D        5:0116:1
                    END                                                                  00024300 D        5:0117:1
                  ELSE BEGIN                % ND = 1-28                                00024400 D        5:0117:1
                  IF JJ LEQ 3 THEN BEGIN NNR:=12; MNR:=47; END                           00024500 D        5:0118:1
                  ELSE IF JJ GEQ 4 AND JJ LEQ 7 THEN BEGIN                               00024600 D        5:0121:2
                     NNR:=19; MNR:=57;                                                   00024700 D        5:0125:1
                     END                                                                 00024800 D        5:0126:3
                  ELSE IF JJ GEQ 8 AND JJ LEQ 11 THEN                                    00024900 D        5:0126:3
                     BEGIN                                                               00025000 D        5:0129:2
                     NNR:=26; MNR:=67;                                                   00025100 D        5:0130:2
                     END                                                                 00025200 D        5:0132:2
                  ELSE IF JJ GEQ 12 AND JJ LEQ 19 THEN                                   00025300 D        5:0132:2
                     BEGIN                                                               00025400 D        5:0135:1
                     NNR:=33; MNR:=77;                                                   00025500 D        5:0136:1
                     END                                                                 00025600 D        5:0138:1
                  ELSE IF JJ GEQ 20 THEN                                                 00025700 D        5:0138:1
                     BEGIN                                                               00025800 D        5:0140:0
                     NNR:=40;                                                            00025900 D        5:0141:0
                     MNR:=87;                                                            00026000 D        5:0141:3
                     END;                                                                00026100 D        5:0143:0
               TR(6);                                                                    00026200 D        5:0143:0
                     IF ST[IND,3]=2 THEN                                                 00026300 D        5:0143:0
                        BEGIN                                                            00026400 D        5:0145:2
               TR(7);                                                                    00026500 D        5:0146:2
                        FOR J:=0 STEP 1 UNTIL 1 DO                                       00026600 D        5:0146:2
                           BEGIN                                                         00026700 D        5:0151:1
                           NR:=NNR+J;                                                    00026800 D        5:0151:1
                           READ (INFO[NR],MOUT);                                         00026900 D        5:0152:3
                           WRITE(MSG[J+44],MOUT);                                        00027000 D        5:0155:3
                           END;                                                          00027100 D        5:0159:2
                        REPLACE PMT BY HEAD1,HEAD2;                                      00027200 D        5:0160:0
                        WRITE(MSG[46],MOUT);                                             00027300 D        5:0164:1
                        FOR J:=2 STEP 1 UNTIL 6 DO                                       00027400 D        5:0167:1
                           BEGIN                                                         00027500 D        5:0172:0
                           NR:=NNR+J;                                                    00027600 D        5:0172:0
                           READ (INFO[NR],MOUT);                                         00027700 D        5:0173:2
                           WRITE(MSG[J+45],MOUT);                                        00027800 D        5:0176:2
                           END;                                                          00027900 D        5:0180:1
                        NRCNO:= 45;                                                      00028000 D        5:0180:3
                        NOREC := 8;                                                      00028100 D        5:0181:3
                        ST[IND,1] := 2004;                                               00028200 D        5:0182:3
                        END                                                              00028300 D        5:0185:3
                     ELSE IF ST[IND,3]=3 THEN                                            00028400 D        5:0185:3
                        BEGIN                                                            00028500 D        5:0189:1
               TR(8);                                                                    00028600 D        5:0190:1
                        FOR J:=0 STEP 1 UNTIL 1 DO                                       00028700 D        5:0190:1
                           BEGIN                                                         00028800 D        5:0195:0
                           NR:=MNR+J;                                                    00028900 D        5:0195:0
                           READ (INFO[NR],MOUT);                                         00029000 D        5:0196:2
                           WRITE (MSG[J+44],MOUT);                                       00029100 D        5:0199:2
                           END;                                                          00029200 D        5:0203:1
                        REPLACE PMT BY HEAD1,HEAD2;                                      00029300 D        5:0203:3
SEGMENT    2 IS   73 LONG.
                        WRITE(MSG[46],MOUT);                                             00029400 D        5:0208:0
                        FOR J:=2 STEP 1 UNTIL 9 DO                                       00029500 D        5:0211:0
                           BEGIN                                                         00029600 D        5:0215:3
                           NR:=MNR+J;                                                    00029700 D        5:0215:3
                           READ(INFO[NR],MOUT);                                          00029800 D        5:0217:1
                           WRITE(MSG[J+45],MOUT);                                        00029900 D        5:0220:1
                           END;                                                          00030000 D        5:0224:0
                        NRCNO:= 45;                                                      00030100 D        5:0224:2
                        NOREC:=11;                                                       00030200 D        5:0225:2
                        ST[IND,1] := 2004;                                               00030300 D        5:0226:2
                        END                                                              00030400 D        5:0229:2
                     ELSE IF ST[IND,3]=4 THEN                                            00030500 D        5:0229:2
                        BEGIN                                                            00030600 D        5:0229:2
                        READ(INFO[106+JJ],OFIL);                                         00030700 D        5:0233:0
               TR(9);                                                                    00030800 D        5:0234:0
                        FOR J:=0 STEP 1 UNTIL  9 DO                                      00030900 D        5:0237:3
                                                                                         00031000 D        5:0237:3
```

```
                        BEGIN                                                   00031100 D      5:0242:2
                        NR:=97+J;                                               00031200 D      5:0242:2
                          READ(INFO(NR),MOUT);                                  00031300 D      5:0244:2
                    IF J=1 THEN                                                 00031400 D      5:0247:2
                      BEGIN                                                     00031500 D      5:0248:2
                      REPLACE PMT BY POF FOR 8;                                 00031600 D      5:0249:2
                    POF:=POF+8;                                                 00031700 D      5:0251:1
                      REPLACE PMT+33 BY POF FOR 8;                              00031800 D      5:0252:3
                    POF:=POF+8;                                                 00031900 D      5:0255:0
                      REPLACE PMT+56 BY POF FOR 8;                              00032000 D      5:0256:2
                      END                                                       00032100 D      5:0258:3
                    ELSE IF J GEQ 2 AND J LEQ 8 THEN                            00032200 D      5:0258:3
                      BEGIN                                                     00032300 D      5:0262:0
                      POF:=POF+8;                                               00032400 D      5:0263:0
                      REPLACE PMT+56 BY POF FOR 8;                              00032500 D      5:0264:2
                      END;                                                      00032600 D      5:0266:3
                        WRITE (MSG(J+58),MOUT);                                 00032700 D      5:0266:3
                        END;                                                    00032800 D      5:0270:2
                      NRCNO:= 59;                                               00032900 D      5:0271:2
                      NOREC := 10;                                              00033000 D      5:0272:2
                      ST(IND,1):= 2004;                                         00033100 D      5:0273:2
                      END                                                       00033200 D      5:0276:2
                    ELSE BEGIN       % ST(IND,3) NE 2,3,4                     00033300 D      5:0276:2
              TR(10);                                                           00033400 D      5:0277:2
                      FOR J:=0 STEP 1 UNTIL 11 DO                               00033500 D      5:0277:2
                        BEGIN                                                   00033600 D      5:0282:1
                        READ (INFO(J),MOUT);                                    00033700 D      5:0282:1
                        WRITE (MSG(J+33),MOUT);                                 00033800 D      5:0285:2
                        END;                                                    00033900 D      5:0289:1
                      NRCNO:= 34;                                               00034000 D      5:0289:3
                      NOREC := 12;                                              00034100 D      5:0290:3
                      ST(IND,1) := 2004;                                        00034200 D      5:0291:3
                      END;                                                      00034300 D      5:0294:3
                    END;                                                        00034400 D      5:0294:3
                    END;                                                        00034500 D      5:0294:3
                  END;                                                          00034600 D      5:0294:3
                END;                                                            00034700 D      5:0294:3
SEGMENT    5 IS  296 LONG.
        PROCEDURE P4000;                                                        00034800 D      3:0019:3
%                                                                               00034900 D      3:0021:2
%       FILE ACCESS MODE OF OPERATION                                           00035000 D      3:0021:2
%                                                                               00035100 D      3:0021:2
        BEGIN                                                                   00035200 D      3:0021:2
        NEW IFS,MODTYP,LL,L,IADD,IUPDEL;                                        00035300 D      3:0021:3
START OF SEGMENT    7
        POINTER PX,POF;                                                         00035400 D      7:0001:1
        POF:=POINTER(OFIL);                                                     00035500 D      7:0001:3
        IF ST(IND,1)= 4000 THEN                                                 00035600 D      7:0003:2
          BEGIN                                                                 00035700 D      7:0006:1
       TR(41);                                                                  00035800 D      7:0007:1
          SCAN PX:PIC WHILE "0123456789";                                       00035900 D      7:0007:1
          IFS:= INT(PIC,DELTA(PIC,PX));                                         00036000 D      7:0009:3
          IFAC(IND):=IFS;                                                       00036100 D      7:0012:0
          IF IFS LSS 1 OR IFS GEQ 3 THEN                                        00036200 D      7:0013:3
            BEGIN                                                               00036300 D      7:0015:2
            NRCNO:= 11;                                                         00036400 D      7:0016:2
            NOREC:= 1;                                                          00036500 D      7:0017:2
          END                                                                   00036600 D      7:0018:2
        ELSE BEGIN                                                              00036700 D      7:0018:2
          READ(MSG(136),OFIL);                                                  00036800 D      7:0019:2
          IF IFS=1 THEN REPLACE POF+14 BY "4"                                   00036900 D      7:0023:0
          ELSE REPLACE POF+14 BY "6";                                           00037000 D      7:0025:2
          WRITE(MSG(136),OFIL);                                                 00037100 D      7:0029:1
          NRCNO:=137;                                                           00037200 D      7:0032:3
          NOREC:=4;                                                             00037300 D      7:0034:1
          ST(IND,1):=4002;                                                      00037400 D      7:0035:1
          END;                                                                  00037500 D      7:0038:1
        END                                                                     00037600 D      7:0038:1
        ELSE IF ST(IND,1) = 4002 THEN                                           00037700 D      7:0038:1
          BEGIN                                                                 00037800 D      7:0042:0
        TR(42);                                                                 00037900 D      7:0043:0
          SCAN PX:PIC WHILE "0123456789";                                       00038000 D      7:0043:0
          MODTYP:= INT(PIC,DELTA(PIC,PX));                                      00038100 D      7:0045:2
          IFS:= IFAC(IND);                                                      00038200 D      7:0047:3
        TR(MODTYP);                                                             00038300 D      7:0049:1
        TR(IFS);                                                                00038400 D      7:0049:1
        TR(IND);                                                                00038500 D      7:0049:1
        TR(IFAC(IND));                                                          00038600 D      7:0049:1
          IF MODTYP=1 THEN                                                      00038700 D      7:0049:1
            BEGIN                                                               00038800 D      7:0050:0
```

```
      NRCNO:= 158;
      NOREC:=1;
      ST[IND,1]:= 4008;
    END
  ELSE IF MODTYP=2 THEN
    BEGIN
      NRCNO:= 157;
      NOREC:= 1;
      ST[IND,1]:= 4005;
    END
  ELSE BEGIN
      NRCNO:= 11;
      NOREC:=1;
    END;
  END
  ELSE IF ST[IND,1]=4006 THEN
        BEGIN
        NRCNO:=176;
        NOREC:=5;
        ST[IND,1]:=4014;
        END
ELSE IF ST[IND,1]=4005 THEN
  BEGIN
TR(46);
IFS:=IFAC[IND];
IF IFS=1 THEN ST[IND,6]:=FETCH(PIC,4)
ELSE BEGIN
  ST[IND,6]:=FETCH(PIC,4);
  ST[IND,7]:=FETCH(PIC+4,2);
  END;
IF IFS=1 THEN BEGIN
  FOR J:=0 STEP 1 UNTIL 4 DO
    BEGIN
    READ(CDIR[J],OFIL);
    FOR K:=0 STEP 1 UNTIL 19 DO
      BEGIN
      PS:=J*20+K;
      ST[IND,8]:=PS;
      PX:=POINTER(OFIL[K]);
      IF PX="%" THEN
        BEGIN
        NRCNO:=181;
        NOREC:=1;
        ST[IND,1]:=4006;
        K:=K+21;
        J:=J+6;
        ST[IND,8]:=101;
        RETURN;
        END
      ELSE IF ST[IND,6]=OFIL[K] THEN BEGIN
        K:=K+21;
        J:=J+6;
        END;
      END;
    END;
  READ(CKTD[PS],OFIL);
  WRITE(MSG[142],OFIL);
  NRCNO:=141;
  NOREC:=3;
  ST[IND,1]:=4006;
END
ELSE BEGIN
  FOR J:=0 STEP 1 UNTIL 9 DO
    BEGIN
    READ(TDIR[J],OFIL);
    FOR K:=0 STEP 2 UNTIL 18 DO
      BEGIN
      PS:=ROUND(J*10+K/2);
      ST[IND,8]:=PS;
      PX:=POINTER(OFIL[K]);
      IF PX="%" THEN
        BEGIN
        NRCNO:=181;
        NOREC:=1;
        ST[IND,1]:=4006;
        K:=K+21;
        J:=J+6;
        ST[IND,8]:=101;
        RETURN;
        END
```

```
00038900 D    7:0051:0
00039000 D    7:0052:2
00039100 D    7:0053:2
00039200 D    7:0056:2
00039300 D    7:0056:2
00039400 D    7:0058:1
00039500 D    7:0059:1
00039600 D    7:0060:3
00039700 D    7:0061:3
00039800 D    7:0064:3
00039900 D    7:0064:3
00040000 D    7:0065:3
00040100 D    7:0066:3
00040200 D    7:0067:3
00040300 D    7:0067:3
00040400 D    7:0067:3
00040500 D    7:0071:2
00040600 D    7:0072:2
00040700 D    7:0074:0
00040800 D    7:0075:0
00040900 D    7:0078:0
00041000 D    7:0078:0
00041100 D    7:0081:3
00041200 D    7:0082:3
00041300 D    7:0082:3
00041400 D    7:0084:1
00041500 D    7:0089:0
00041600 D    7:0090:2
00041700 D    7:0094:0
00041800 D    7:0098:0
00041900 D    7:0098:0
00042000 D    7:0099:3
00042100 D    7:0104:2
00042200 D    7:0104:2
00042300 D    7:0107:3
00042400 D    7:0112:2
00042500 D    7:0112:2
00042600 D    7:0115:0
00042700 D    7:0118:0
00042800 D    7:0120:0
00042900 D    7:0121:1
00043000 D    7:0122:1
00043100 D    7:0123:3
00043200 D    7:0124:3
00043300 D    7:0127:3
00043400 D    7:0129:2
00043500 D    7:0131:1
00043600 D    7:0134:2
00043700 D    7:0134:3
00043800 D    7:0134:3
00043900 D    7:0140:0
00044000 D    7:0141:3
00044100 D    7:0143:2
00044200 D    7:0143:2
00044300 D    7:0144:2
00044400 D    7:0145:2
00044500 D    7:0148:3
00044600 D    7:0152:1
00044700 D    7:0153:3
00044800 D    7:0154:3
00044900 D    7:0157:3
00045000 D    7:0157:3
00045100 D    7:0158:3
00045200 D    7:0163:2
00045300 D    7:0163:2
00045400 D    7:0166:3
00045500 D    7:0171:3
00045600 D    7:0171:3
00045700 D    7:0175:0
00045800 D    7:0178:0
00045900 D    7:0180:0
00046000 D    7:0181:1
00046100 D    7:0182:1
00046200 D    7:0183:3
00046300 D    7:0184:3
00046400 D    7:0187:3
00046500 D    7:0189:2
00046600 D    7:0191:1
00046700 D    7:0194:2
00046800 D    7:0194:3
```

```
            ELSE IF ST[IND,6]=OFIL[K] AND ST[IND,7]=OFIL[K+1]
            THEN BEGIN
               K:=K+21;
               J:=J+11;
               END;
            END;
         END;
      READ(TNKD[PS],OFIL);
      WRITE(MSG[145],OFIL);
      NRCNO:=144;
      NOREC:=3;
      ST[IND,1]:=4006;
   END;
   END
   ELSE IF ST[IND,1]=4008 THEN
      BEGIN
   IFS:=IFAC[IND];
   IF IFS=1 THEN ST[IND,6]:=FETCH(PIC,4)
   ELSE BEGIN
      ST[IND,6]:=FETCH(PIC,4);
      ST[IND,7]:=FETCH(PIC+4,2);
      END;
   IF IFS=1 THEN BEGIN
      FOR J:=0 STEP 1 UNTIL 4 DO
         BEGIN
         ST[IND,9]:=J;
         READ(CDIR[J],OFIL);
         FOR K:=0 STEP 1 UNTIL 19 DO
            BEGIN
            ST[IND,5]:=K;
            PS:=J*20+K;
            ST[IND,8]:=PS;
            PX:=POINTER(OFIL[K]);
            IF PX="%" THEN
               BEGIN
               NRCNO:=166;
               NOREC:=4;
               ST[IND,1]:=4009;
               K:=K+21;
               J:=J+6;
               ST[IND,8]:=101;
               RETURN;
               END
            ELSE IF ST[IND,6]=OFIL[K] THEN BEGIN
               K:=K+21;
               J:=J+6;
               END;
            END;
         END;
      NRCNO:=160;
      NOREC:=3;
      ST[IND,1]:=4011;
   END
   ELSE BEGIN
      FOR J:=0 STEP 1 UNTIL 9 DO
         BEGIN
         ST[IND,9]:=J;
         READ(TDIR[J],OFIL);
         FOR K:=0 STEP 2 UNTIL 18 DO
            BEGIN
            ST[IND,5]:=K;
            PS:=ROUND(J*10+K/2);
            ST[IND,8]:=PS;
            PX:=POINTER(OFIL[K]);
            IF PX="%" THEN
               BEGIN
               NRCNO:=166;
               NOREC:=4;
               ST[IND,1]:=4009;
               K:=K+21;
               J:=J+6;
               ST[IND,8]:=101;
               RETURN;
               END
            ELSE IF ST[IND,6]=OFIL[K] AND ST[IND,7]=OFIL[K+1]
            THEN BEGIN
               K:=K+21;
               J:=J+11;
               END;
            END;
```

```
00046900 D  7:0194:3
00047000 D  7:0201:3
00047100 D  7:0203:3
00047200 D  7:0205:2
00047300 D  7:0207:1
00047400 D  7:0207:1
00047500 D  7:0208:1
00047600 D  7:0209:1
00047700 D  7:0212:2
00047800 D  7:0216:0
00047900 D  7:0217:2
00048000 D  7:0218:2
00048100 D  7:0221:2
00048200 D  7:0221:2
00048300 D  7:0221:2
00048400 D  7:0225:1
00048500 D  7:0226:1
00048600 D  7:0227:3
00048700 D  7:0232:2
00048800 D  7:0234:0
00048900 D  7:0237:2
00049000 D  7:0241:2
00049100 D  7:0241:2
00049200 D  7:0243:1
00049300 D  7:0248:0
00049400 D  7:0248:0
00049500 D  7:0251:0
00049600 D  7:0254:1
00049700 D  7:0259:0
00049800 D  7:0259:0
00049900 D  7:0262:0
00050000 D  7:0264:2
00050100 D  7:0267:2
00050200 D  7:0269:2
00050300 D  7:0270:3
00050400 D  7:0271:3
00050500 D  7:0273:1
00050600 D  7:0274:1
00050700 D  7:0277:1
00050800 D  7:0279:0
00050900 D  7:0280:3
00051000 D  7:0284:0
00051100 D  7:0284:1
00051200 D  7:0284:1
00051300 D  7:0289:2
00051400 D  7:0291:1
00051500 D  7:0293:0
00051600 D  7:0293:0
00051700 D  7:0294:0
00051800 D  7:0295:0
00051900 D  7:0296:2
00052000 D  7:0297:2
00052100 D  7:0300:2
00052200 D  7:0300:2
00052300 D  7:0301:2
00052400 D  7:0306:1
00052500 D  7:0306:1
00052600 D  7:0309:1
00052700 D  7:0312:2
00052800 D  7:0317:2
00052900 D  7:0317:2
00053000 D  7:0320:2
00053100 D  7:0323:3
00053200 D  7:0326:3
00053300 D  7:0328:3
00053400 D  7:0330:0
00053500 D  7:0331:0
00053600 D  7:0332:2
00053700 D  7:0333:2
00053800 D  7:0336:2
00053900 D  7:0338:1
00054000 D  7:0340:0
00054100 D  7:0343:1
00054200 D  7:0343:2
00054300 D  7:0343:2
00054400 D  7:0350:2
00054500 D  7:0352:2
00054600 D  7:0354:1
00054700 D  7:0356:0
00054800 D  7:0356:0
```

```
      END;
   NRCNO:=160;
   NOREC:=3;
   ST[IND,1]:=4011;
   END;
     END
ELSE IF ST[IND,1]=4009 THEN
  BEGIN
TR(51);
   SCAN PX:PIC WHILE "0123456789";
   IADD:= INT(PIC,DELTA(PIC,PX));
   IFS:= IFAC[IND];
   IF IADD=2 THEN
     BEGIN
      NRCNO:= 175;
      NOREC:= 5;
      ST[IND,1]:= 4014;
     END
  ELSE BEGIN
   READ(MSG[136],OFIL);
   IF IFS=1 THEN REPLACE POF+14 BY "4"
   ELSE REPLACE POF+14 BY "6";
   WRITE(MSG[33],OFIL);
   READ(MSG[171],OFIL);
   WRITE(MSG[34],OFIL);
   READ(MSG[172],OFIL);
   WRITE(MSG[35],OFIL);
   IF IFS=1 THEN LL:=0
   ELSE LL:=3;
   READ(MSG[140+LL],OFIL);
   WRITE(MSG[36],OFIL);
   READ(MSG[141+LL],OFIL);
   WRITE(MSG[37],OFIL);
   REPLACE POF BY US,(" ") FOR 78,RS;
   WRITE(MSG[38],OFIL);
   FORMF:=1;
    NRCNO:= 34;
    NOREC:= 6;
    ST[IND,1]:= 4010;
    END;
 END
ELSE IF ST[IND,1]= 4010 THEN
  BEGIN
TR(52);
   IFS:= IFAC[IND];
   REPLACE POF BY PIC FOR 78,(" ") FOR 2;
IF IFS=1 THEN BEGIN
   FOR J:=0 STEP 1 UNTIL 4 DO
      BEGIN
      READ(CDIR[J],MOUT);
      FOR K:=0 STEP 1 UNTIL 19 DO
         BEGIN
         PS:=J*20+K;
         ST[IND,8]:=PS;
         PX:=POINTER(MOUT[K]);
         IF PX="     " THEN
            BEGIN
            MOUT[K]:=ST[IND,6];
            WRITE(CDIR[J],MOUT);
            K:=K+21;
            J:=J+6;
            END
         ELSE
         IF PX="%" THEN
            BEGIN
            MOUT[K]:=ST[IND,6];
            IF K=19 THEN
                BEGIN
                WRITE(CDIR[J],MOUT);
                READ(CDIR[J+1],MOUT);
                PX:=POINTER(MOUT);
                REPLACE PX BY "%";
                WRITE(CDIR[J+1],MOUT);
                END
            ELSE BEGIN
                REPLACE PX+4 BY "%";
                WRITE(CDIR[J],MOUT);
                END;
            K:=K+21;
            J:=J+6;
```

```
00054900 D        7:0357:0
00055000 D        7:0358:0
00055100 D        7:0359:2
00055200 D        7:0360:2
00055300 D        7:0363:2
00055400 D        7:0363:2
00055500 D        7:0363:2
00055600 D        7:0367:1
00055700 D        7:0368:1
00055800 D        7:0368:1
00055900 D        7:0370:3
00056000 D        7:0373:0
00056100 D        7:0374:2
00056200 D        7:0375:1
00056300 D        7:0376:1
00056400 D        7:0377:3
00056500 D        7:0378:3
00056600 D        7:0381:3
00056700 D        7:0381:3
00056800 D        7:0382:3
00056900 D        7:0386:1
00057000 D        7:0388:3
00057100 D        7:0392:2
00057200 D        7:0395:2
00057300 D        7:0399:0
00057400 D        7:0402:0
00057500 D        7:0405:2
00057600 D        7:0408:2
00057700 D        7:0410:2
00057800 D        7:0412:3
00057900 D        7:0416:3
00058000 D        7:0419:3
00058100 D        7:0423:3
00058200 D        7:0426:3
00058300 D        7:0431:0
00058400 D        7:0434:0
00058500 D        7:0435:0
00058600 D        7:0436:0
00058700 D        7:0437:0
00058800 D        7:0440:0
00058900 D        7:0440:0
00059000 D        7:0440:0
00059100 D        7:0443:3
00059200 D        7:0444:3
00059300 D        7:0444:3
00059400 D        7:0446:1
00059500 D        7:0449:1
00059600 D        7:0451:0
00059700 D        7:0455:3
00059800 D        7:0455:3
00059900 D        7:0459:0
00060000 D        7:0463:3
00060100 D        7:0463:3
00060200 D        7:0466:1
00060300 D        7:0469:1
00060400 D        7:0471:1
00060500 D        7:0473:0
00060600 D        7:0474:0
00060700 D        7:0477:2
00060800 D        7:0480:3
00060900 D        7:0482:2
00061000 D        7:0484:1
00061100 D        7:0484:1
00061200 D        7:0485:1
00061300 D        7:0486:2
00061400 D        7:0487:2
00061500 D        7:0491:0
00061600 D        7:0492:0
00061700 D        7:0493:0
00061800 D        7:0496:1
00061900 D        7:0499:3
00062000 D        7:0501:2
00062100 D        7:0502:3
00062200 D        7:0506:1
00062300 D        7:0506:1
00062400 D        7:0507:1
00062500 D        7:0509:0
00062600 D        7:0512:1
00062700 D        7:0512:1
00062800 D        7:0514:0
```

```
            END;
         END;
      END;
   WRITE(CKTD[PS],OFIL);
   END
ELSE BEGIN
   FOR J:=0 STEP 1 UNTIL 9 DO
      BEGIN
      READ(TDIR[J],MOUT);
      FOR K:=0 STEP 2 UNTIL 18 DO
         BEGIN
         PS:=ROUND(J*10+K/2);
         ST[IND,8]:=PS;
         PX:=POINTER(MOUT[K]);
         IF PX="    " THEN
            BEGIN
            MOUT[K]:=ST[IND,6];
            MOUT[K+1]:=ST[IND,7];
            WRITE(TDIR[J],MOUT);
            K:=K+2;
            J:=J+6;
            END
         ELSE
         IF PX="%" THEN
            BEGIN
            MOUT[K]:=ST[IND,6];
            MOUT[K+1]:=ST[IND,7];
            IF K=18 THEN
                BEGIN
                WRITE(TDIR[J],MOUT);
                READ(TDIR[J+1],MOUT);
                PX:=POINTER(MOUT);
                REPLACE PX BY "%";
                WRITE(TDIR[J+1],MOUT);
                END
            ELSE BEGIN
                REPLACE PX+8 BY "%";
                WRITE(TDIR[J],MOUT);
                END;
            K:=K+2;
            J:=J+11;
            END;
         END;
      END;
   WRITE(TNKD[PS],OFIL);
   END;
   NRCNO:= 175;
   NOREC:= 6;
   ST[IND,1]:= 4014;
  END
ELSE IF ST[IND,1]= 4011 THEN
  BEGIN
TR(53);
   IUPDEL:=INT(PIC,1);
   IFS:= IFAC[IND];
   ST[IND,1]:= 4012;
   IF IUPDEL=2 THEN
    BEGIN
      IF IFS=1 THEN BEGIN
         L:=ST[IND,9];              % J INDEx
         READ(CDIR[L],OFIL);
         LL:=ST[IND,5];             % K INDEx
         PX:=POINTER(OFIL[LL]);
         REPLACE PX BY (" ") FOR 4;
         WRITE(CDIR[L],OFIL);
         ST[IND,8]:=101;
         END
      ELSE BEGIN
         L:=ST[IND,9];
         READ(TDIR[L],OFIL);
         LL:=ST[IND,5];
         PX:=POINTER(OFIL[LL]);
         REPLACE PX BY (" ") FOR 8;
         WRITE(TDIR[L],OFIL);
         ST[IND,8]:=101;
         END;
      NRCNO:= 175;
      NOREC:= 6;
      ST[IND,1]:= 4014;
    END
```

```
00062900 D        7:0515:3
00063000 D        7:0515:3
00063100 D        7:0516:3
00063200 D        7:0517:3
00063300 D        7:0521:0
00063400 D        7:0521:0
00063500 D        7:0522:0
00063600 D        7:0526:3
00063700 D        7:0526:3
00063800 D        7:0530:0
00063900 D        7:0535:0
00064000 D        7:0535:0
00064100 D        7:0538:1
00064200 D        7:0541:1
00064300 D        7:0543:1
00064400 D        7:0545:0
00064500 D        7:0546:0
00064600 D        7:0549:2
00064700 D        7:0553:1
00064800 D        7:0556:2
00064900 D        7:0558:1
00065000 D        7:0560:0
00065100 D        7:0560:0
00065200 D        7:0561:0
00065300 D        7:0562:1
00065400 D        7:0563:1
00065500 D        7:0566:3
00065600 D        7:0570:2
00065700 D        7:0571:2
00065800 D        7:0572:2
00065900 D        7:0575:3
00066000 D        7:0579:1
00066100 D        7:0581:0
00066200 D        7:0582:1
00066300 D        7:0585:3
00066400 D        7:0585:3
00066500 D        7:0586:3
00066600 D        7:0588:2
00066700 D        7:0591:3
00066800 D        7:0591:3
00066900 D        7:0593:2
00067000 D        7:0595:1
00067100 D        7:0595:1
00067200 D        7:0596:1
00067300 D        7:0597:1
00067400 D        7:0600:2
00067500 D        7:0600:2
00067600 D        7:0602:0
00067700 D        7:0603:0
00067800 D        7:0606:0
00067900 D        7:0606:0
00068000 D        7:0609:3
00068100 D        7:0610:3
00068200 D        7:0610:3
00068300 D        7:0612:1
00068400 D        7:0613:3
00068500 D        7:0616:3
00068600 D        7:0617:2
00068700 D        7:0618:2
00068800 D        7:0620:1
00068900 D        7:0622:3
00069000 D        7:0625:3
00069100 D        7:0628:1
00069200 D        7:0630:0
00069300 D        7:0631:1
00069400 D        7:0634:1
00069500 D        7:0637:2
00069600 D        7:0637:2
00069700 D        7:0638:2
00069800 D        7:0641:0
00069900 D        7:0644:0
00070000 D        7:0646:2
00070100 D        7:0648:1
00070200 D        7:0649:2
00070300 D        7:0652:2
00070400 D        7:0655:3
00070500 D        7:0655:3
00070600 D        7:0657:1
00070700 D        7:0658:1
00070800 D        7:0661:1
```

```
   ELSE BEGIN                    %IUPDEL NEQ 2
      PS:=ST[IND,8];
      IF IFS=1 THEN READ(CKTD[PS],MOUT)
      ELSE READ(TNKD[PS],MOUT);
      PMT:=POINTER(MOUT);
   READ(MSG[136],OFIL);
   IF IFS=1 THEN REPLACE POF+14 BY "4"
   ELSE REPLACE POF+14 BY "6";
   WRITE(MSG[33],OFIL);
   READ(MSG[171],OFIL);
   WRITE(MSG[34],OFIL);
   READ(MSG[172],OFIL);
   WRITE(MSG[35],OFIL);
   IF IFS=1 THEN LL:=0
   ELSE LL:=3;
   READ(MSG[140+LL],OFIL);
   WRITE(MSG[36],OFIL);
   READ(MSG[141+LL],OFIL);
   WRITE(MSG[37],OFIL);
   REPLACE POF BY US,PMT FOR 78,RS;
   WRITE(MSG[38],OFIL);
   FORMF:=1;
    NRCNO:= 34;
    NOREC:= 6;
    END;
 END
ELSE IF ST[IND,1]= 4012 THEN
  BEGIN
TR(54);
   IFS:= IFAC[IND];
   REPLACE POF BY PIC FOR 78,(" ") FOR 2;
   PS:= ST[IND,8];
   IF IFS=1 THEN
   WRITE(CKTD[PS],OFIL)
   ELSE WRITE(TNKD[PS],OFIL);
   NRCNO:=175;
   NOREC:= 6;
   ST[IND,1]:= 4014;
  END
ELSE IF ST[IND,1]= 4014 THEN
  BEGIN
TR(55);
   I:=INT(PIC,1);
   IF I=1 THEN
     BEGIN
      NRCNO:= 138;
      NOREC:=3;
      ST[IND,1]:= 4002;
     END
   ELSE IF I=2 THEN
     BEGIN
      NRCNO:= 128;
      NOREC:= 3;
      ST[IND,1]:= 4000;
     END
   ELSE IF I=3 THEN
     BEGIN
      NRCNO:= 5;
      NOREC:=5;
      ST[IND,1]:=0004;
     END
   ELSE IF I=4 THEN
     BEGIN
      NRCNO:= 21;
      NOREC:=1;
      ST[IND,1]:=9999;
     END
   ELSE IF I=5 THEN
     BEGIN
      IFS:= IFAC[IND];
      IF ST[IND,8]=101 THEN
         BEGIN
         NRCNO:=11;
         NOREC:=1;
         END
      ELSE IF IFS=1 THEN
         BEGIN
         READ(CKTD[ST[IND,8]],OFIL);
         WRITE(MSG[142],OFIL);
         NRCNO:=141;
```

```
00070900 D        7:0661:1
00071000 D        7:0662:1
00071100 D        7:0665:0
00071200 D        7:0669:2
00071300 D        7:0674:1
00071400 D        7:0676:1
00071500 D        7:0679:3
00071600 D        7:0682:1
00071700 D        7:0686:0
00071800 D        7:0689:0
00071900 D        7:0692:2
00072000 D        7:0695:2
00072100 D        7:0699:0
00072200 D        7:0702:0
00072300 D        7:0704:0
00072400 D        7:0706:1
00072500 D        7:0710:1
00072600 D        7:0713:1
00072700 D        7:0717:1
00072800 D        7:0720:1
00072900 D        7:0725:0
00073000 D        7:0728:0
00073100 D        7:0729:0
00073200 D        7:0730:0
00073300 D        7:0731:0
00073400 D        7:0731:0
00073500 D        7:0731:0
00073600 D        7:0734:3
00073700 D        7:0735:3
00073800 D        7:0735:3
00073900 D        7:0737:1
00074000 D        7:0740:1
00074100 D        7:0743:0
00074200 D        7:0743:3
00074300 D        7:0747:2
00074400 D        7:0752:1
00074500 D        7:0753:3
00074600 D        7:0754:3
00074700 D        7:0757:3
00074800 D        7:0757:3
00074900 D        7:0761:2
00075000 D        7:0762:2
00075100 D        7:0762:2
00075200 D        7:0764:1
00075300 D        7:0765:1
00075400 D        7:0766:1
00075500 D        7:0767:3
00075600 D        7:0768:3
00075700 D        7:0771:3
00075800 D        7:0771:3
00075900 D        7:0773:3
00076000 D        7:0774:3
00076100 D        7:0776:1
00076200 D        7:0777:1
00076300 D        7:0780:1
00076400 D        7:0780:1
00076500 D        7:0782:1
00076600 D        7:0783:1
00076700 D        7:0784:1
00076800 D        7:0785:1
00076900 D        7:0787:3
00077000 D        7:0787:3
00077100 D        7:0789:3
00077200 D        7:0790:3
00077300 D        7:0791:3
00077400 D        7:0792:3
00077500 D        7:0795:3
00077600 D        7:0795:3
00077700 D        7:0797:3
00077800 D        7:0798:3
00077900 D        7:0800:1
00078000 D        7:0803:1
00078100 D        7:0804:1
00078200 D        7:0805:1
00078300 D        7:0806:1
00078400 D        7:0806:1
00078500 D        7:0808:0
00078600 D        7:0809:0
00078700 D        7:0813:3
00078800 D        7:0817:1
```

```
                    NOREC:=3;
                    ST[IND,1]:=4006;
                    END
                ELSE BEGIN
                    READ(INKD[ST[IND,8]],OFIL);
                    WRITE(MSG[145],OFIL);
                    NRCNO:=144;
                    NOREC:=3;
                    ST[IND,1]:=4006;
                    END;
              END
              ELSE BEGIN      % INVALID ENTRY- NOT 1-5
                NRCNO:= 11;
                NOREC:=1;
              END;
            END
            ELSE RETURN;
            END;
SEGMENT    7 IS  843 LONG.

            PROCEDURE MST;
              BEGIN END;

          %  PROCEDURE DIVISION- MAIN PROGRAM- P0000

          %   INITIALIZATION
              FOR IND:=0 STEP 1 UNTIL 3 DO
                 BEGIN
                 MSGNO[IND]:=0;
                 ST[IND,1]:=2;
                 END;
              FORMF:=0;
              PMT:= POINTER (MOUT);     %POINTING TO MOUT[0,0]
              PCN  := POINTER(ICON);
              PICH:=POINTER(ICODE);
              PIC := POINTER(ICODE);
              REPLACE POINTER(SET) BY 4"FF" FOR 8;
              REPLACE PIC BY CLEAR,LF,
                 "THIS IS THE ESM",CR,LF,
                 "ENTER USERCODE PLEASE",HOME,ETX;
              WRITE(CRTX,ICODE);
              WHILE READ (CRTX,PIC) DO
              BEGIN
                 TR(11);
              SCAN P1:PIC WHILE " ";
              ICX[0]:=4"10041008";
              ICX[1]:=4"10121019";
              SCAN PICH:PIC WHILE SET;
              PICH:=PICH+1;
              BADH:=0;
              IF  PICH+4=IC4H THEN BEGIN
                 IND:=0;
                 REPLACE PICH+4 BY IWC4;
                 END
              ELSE IF PICH+4=IC8H THEN BEGIN
                 IND:=1;
                 REPLACE PICH+4 BY IWC8;
                 END
              ELSE IF PICH+4=IC18H THEN BEGIN
                 IND:=2;
                 REPLACE PICH+4 BY IWC18;
                 END
              ELSE  BEGIN
                 IND:=3;
            TR(12);
                 REPLACE PICH+4 BY IWC25;
                 END;

              IF BADH=1 THEN BEGIN TR(13);END
            ELSE IF PICH=POINTER(MSGNO[IND])+3 FOR 2 AND IND NEQ 3 THEN BEGIN END
              ELSE BEGIN
              REPLACE POINTER(MSGNO[IND])+3 BY PICH FOR 2;
             TR(14);
              IF P1="ABORT" THEN BEGIN
                 REPLACE PIC BY HOME,ETX;
              IF IND=3 THEN REPLACE PIC BY CR,LF,ETX;
                 WRITE(CRTX,ICODE);
                 RETURN;
                 END;
              IF P1="ATTACH" THEN BEGIN
```

```
00078900 D        7:0818:3
00079000 D        7:0819:3
00079100 D        7:0822:3
00079200 D        7:0822:3
00079300 D        7:0823:3
00079400 D        7:0828:2
00079500 D        7:0832:0
00079600 D        7:0833:2
00079700 D        7:0834:2
00079800 D        7:0837:2
00079900 D        7:0837:2
00080000 D        7:0837:2
00080100 D        7:0838:2
00080200 D        7:0839:2
00080300 D        7:0840:2
00080400 D        7:0840:2
00080500 D        7:0840:2
00080600 D        7:0841:3

00080700 D        3:0023:0
00080800 D        3:0023:0
00080900 D        3:0024:3
00081000 D        3:0025:1
00081100 D        3:0025:1
00081200 D        3:0025:1
00081300 D        3:0025:1
00081400 D        3:0025:1
00081500 D        3:0030:0
00081600 D        3:0030:0
00081700 D        3:0031:2
00081800 D        3:0033:3
00081900 D        3:0034:1
00082000 D        3:0035:1
00082100 D        3:0036:3
00082200 D        3:0038:1
00082300 D        3:0040:0
00082400 D        3:0041:2
00082500 D        3:0043:3
00082600 D        3:0047:3
00082700 D        3:0053:1
00082800 D        3:0057:3
00082900 D        3:0060:3
00083000 D        3:0063:1
00083100 D        3:0064:1
00083200 D        3:0064:1
00083300 D        3:0066:2
00083400 D        3:0068:1
00083500 D        3:0070:0
00083600 D        3:0072:0
00083700 D        3:0074:0
00083800 D        3:0075:0
00083900 D        3:0078:1
00084000 D        3:0079:1
00084100 D        3:0082:0
00084200 D        3:0082:0
00084300 D        3:0086:3
00084400 D        3:0087:3
00084500 D        3:0090:2
00084600 D        3:0090:2
00084700 D        3:0095:1
00084800 D        3:0096:1
00084900 D        3:0099:0
00085000 D        3:0099:0
00085100 D        3:0100:0
00085110 D        3:0101:0
00085200 D        3:0101:0
00085300 D        3:0103:3
00085400 D        3:0103:3
00085500 D        3:0103:3
00085600 D        3:0105:3
00085700 D        3:0111:2
00085800 D        3:0112:2
00085810 D        3:0115:2
00085900 D        3:0115:2
00086000 D        3:0118:1
00086050 D        3:0121:1
00086100 D        3:0128:3
00086200 D        3:0131:3
00086300 D        3:0132:0
00086400 D        3:0132:0
```

```
REPLACE PCN BY LEOP,ICVA,4"40" FOR 1,4"00" FOR 1;                    00086500 D      3:0134:3
REPLACE PCN+6 BY PICH+4 FOR 2;                                       00086600 D      3:0141:0
I:=INT(P1+6,2);                                                      00086700 D      3:0142:3
P3:=POINTER(I);                                                      00086800 D      3:0145:3
REPLACE PCN+8 BY P3+3 FOR 1,ETX,LEOP;                                00086900 D      3:0147:1
ICFLG:=1;                                                            00087000 D      3:0152:2
NRCNO:=21;                                                           00087100 D      3:0153:2
NOREC:=1;                                                            00087200 D      3:0154:2
ST[IND,1]:=9999;                                                     00087300 D      3:0155:2
END ELSE                                                             00087400 D      3:0158:1
IF P1= "DS" THEN                                                     00087500 D      3:0158:1
  BEGIN                                                              00087600 D      3:0161:0
   TR(12);                                                           00087700 D      3:0162:0
  NRCNO := 21;                                                       00087800 D      3:0162:0
  NOREC := 1;                                                        00087900 D      3:0163:0
  ST[IND,1] := 9999;                                                 00088000 D      3:0164:0
  END                                                                00088100 D      3:0166:3
ELSE BEGIN                                                           00088200 D      3:0166:3
   TR(13);                                                           00088300 D      3:0167:3
  IF ST[IND,1]= 0001 THEN                                            00088400 D      3:0167:3
    BEGIN                                                            00088500 D      3:0169:3
    NRCNO:= 1;                                                       00088600 D      3:0170:3
    NOREC:= 2;                                                       00088700 D      3:0171:3
    ST[IND,1]:= 0002;                                                00088800 D      3:0172:3
    END                                                              00088900 D      3:0175:0
  ELSE IF ST[IND,1]= 0002 THEN                                       00089000 D      3:0175:0
    BEGIN                                                            00089100 D      3:0178:0
   TR(14);                                                           00089200 D      3:0179:0
   IF P1="OWEN" OR P1="DAN" THEN BEGIN                               00089300 D      3:0179:0
    NRCNO:= 3;                                                       00089400 D      3:0183:3
    NOREC:=1;                                                        00089500 D      3:0184:3
    ST[IND,1]:= 0003;                                                00089600 D      3:0185:3
   END ELSE BEGIN                                                    00089700 D      3:0188:0
      NRCNO:=21;                                                     00089800 D      3:0189:0
      NOREC:=1;                                                      00089900 D      3:0190:0
      ST[IND,1]:=9999;                                               00090000 D      3:0191:0
      END;                                                           00090100 D      3:0193:3
    END                                                              00090200 D      3:0193:3
  ELSE IF ST[IND,1]= 0003 THEN                                       00090300 D      3:0193:3
    BEGIN                                                            00090400 D      3:0196:3
   TR(15);                                                           00090500 D      3:0197:3
   IF P1="ESM" THEN BEGIN                                            00090600 D      3:0197:3
    NRCNO:= 4;                                                       00090700 D      3:0200:2
    NOREC:= 6;                                                       00090800 D      3:0201:2
    ST[IND,1]:= 0004;                                                00090900 D      3:0202:2
   END ELSE BEGIN                                                    00091000 D      3:0204:3
      NRCNO:=21;                                                     00091100 D      3:0205:3
      NOREC:=1;                                                      00091200 D      3:0206:3
      ST[IND,1]:=9999;                                               00091300 D      3:0207:3
      END;                                                           00091400 D      3:0210:2
    END                                                              00091500 D      3:0210:2
  ELSE IF ST[IND,1]= 0004 THEN                                       00091600 D      3:0210:2
    BEGIN                                                            00091700 D      3:0213:2
   TR(16);                                                           00091800 D      3:0214:2
    I:= INT( P1,1);                                                  00091900 D      3:0214:2
    IF I=1 THEN                                                      00092000 D      3:0216:0
      BEGIN                                                          00092100 D      3:0217:0
      NRCNO:= 12;                                                    00092200 D      3:0218:0
      NOREC:= 2;                                                     00092300 D      3:0219:0
      ST[IND,1] := 1000;                                             00092400 D      3:0220:0
      END                                                            00092500 D      3:0222:3
    ELSE IF I=2 THEN                                                 00092600 D      3:0222:3
      BEGIN                                                          00092700 D      3:0224:3
      NRCNO := 22;                                                   00092800 D      3:0225:3
      NOREC:= 5;                                                     00092900 D      3:0226:3
      ST[IND,1] := 2000;                                             00093000 D      3:0227:3
      END                                                            00093100 D      3:0230:2
    ELSE IF I=4 THEN                                                 00093200 D      3:0230:2
      BEGIN                                                          00093300 D      3:0232:2
      NRCNO:= 128;                                                   00093400 D      3:0233:2
      NOREC:= 3;                                                     00093500 D      3:0235:0
      ST[IND,1]:= 4000;                                              00093600 D      3:0236:0
      END                                                            00093700 D      3:0238:3
    ELSE BEGIN                                                       00093800 D      3:0238:3
      NRCNO := 11;                                                   00093900 D      3:0239:3
      NOREC:= 1;                                                     00094000 D      3:0240:3
      END;                                                           00094100 D      3:0241:3
    END                                                              00094200 D      3:0241:3
    ELSE IF ST[IND,1] GEQ 1000 AND ST[IND,1] LSS 1999 THEN P1000     00094300 D      3:0241:3
    ELSE IF ST[IND,1] GEQ 2000 AND ST[IND,1] LSS 2999 THEN P2000     00094400 D      3:0249:0
```

```
                ELSE IF ST[IND,1] GEQ 4000 AND ST[IND,1] LSS 4999 THEN P4000      00094500 D        3:0257:1
                ELSE BEGIN                                                        00094600 D        3:0265:2
               TR(17);                                                            00094700 D        3:0267:2
                  NRCNO:= 1;                                                      00094800 D        3:0267:2
                  NOREC := 2;                                                     00094900 D        3:0268:2
                  ST[IND,1]:= 0002;                                               00095000 D        3:0269:2
                  END;                                                            00095100 D        3:0271:3
                END;                                                              00095200 D        3:0271:3
%                                                                                 00095300 D        3:0271:3
%-----WRITE TO LOOP-----%                                                         00095400 D        3:0271:3
%                                                                                 00095500 D        3:0271:3
    IF ST[IND,1] = 9999 THEN ST[1,1] := 1;                                        00095600 D        3:0271:3
    NRCNO:= NRCNO - 1;                                                            00095700 D        3:0276:2
    REPLACE PIC BY (" ") FOR 256;                                                 00095800 D        3:0278:0
      TR(18);                                                                     00095900 D        3:0279:3
    IF NRCNO = 10 THEN                                                            00096000 D        3:0279:3
      BEGIN                                                                       00096100 D        3:0280:3
      READ (MSG[NRCNO], MOUT);                                                    00096200 D        3:0281:3
      REPLACE PIC BY HOME, 4"4A" FOR 12,                                          00096300 D        3:0284:3
          PMT FOR 80, HOME, ETX;                                                  00096400 D        3:0287:2
     IF IND=3 THEN REPLACE PIC BY CR,LF,PMT FOR 80,CR,LF,ETX;                     00096410          3:0291:3
      WRITE (CRTX,ICODE);                                                         00096500 D        3:0304:2
      END                                                                         00096600 D        3:0307:2
      ELSE BEGIN                                                                  00096700 D        3:0307:2
        REPLACE P1:PIC BY CLEAR,LF;                                               00096800 D        3:0308:2
    IF IND=3 THEN REPLACE P1:PIC BY LEOP,SEQNO,ZER,IHC25,CR,LF;                   00096810          3:0313:1
        FOR J:=1 STEP 1 UNTIL NOREC DO                                            00096900 D        3:0327:1
            BEGIN                                                                 00097000 D        3:0332:1
            NN:=NRCNO+J-1;                                                        00097100 D        3:0332:1
            READ(MSG[NN],MOUT);                                                   00097200 D        3:0334:2
            P3:=PMT+80;                                                           00097300 D        3:0337:2
            WHILE (P3:=P3-2)="  " DO  BEGIN END;                                  00097400 D        3:0339:3
            P3:=P3+2;                                                             00097500 D        3:0344:3
            IF (DELTA(PIC,P1)+DELTA(PMT,P3)) GEQ 249 THEN                         00097600 D        3:0346:3
              BEGIN                                                               00097700 D        3:0349:3
%           WRITE PACKET TO LOOP                                                  00097800 D        3:0350:3
              REPLACE P1:P1 BY STAYREC,ETX;                                       00097900 D        3:0350:3
              WRITE(CRTX,ICODE);                                                  00098000 D        3:0354:1
              P1:=PIC;                                                            00098100 D        3:0357:1
              END;                                                                00098200 D        3:0358:1
%           LOAD ICODE                                                            00098300 D        3:0358:1
              REPLACE P1:P1 BY PMT FOR DELTA(PMT,P3),CR,LF;                       00098400 D        3:0358:1
            END;                                                                  00098500 D        3:0364:3
%     WRITE LAST PACKET                                                           00098600 D        3:0365:3
        IF FORMF=1 THEN                                                           00098700 D        3:0365:3
          BEGIN                                                                   00098800 D        3:0366:3
          REPLACE P1:P1 BY SETFORMS,HOME,ETX;                                     00098900 D        3:0367:3
    IF IND=3 THEN REPLACE P1:P1 BY CR,LF,ETX;                                     00098910          3:0373:3
          FORMF:=0;                                                               00099000 D        3:0381:3
          END                                                                     00099100 D        3:0382:3
        ELSE                                                                      00099200 D        3:0382:3
          REPLACE P1:P1 BY HOME,ETX;                                              00099300 D        3:0383:3
    IF IND=3 THEN REPLACE P1:P1 BY CR,LF,ETX;                                     00099310          3:0387:1
          WRITE(CRTX,ICODE);                                                      00099400 D        3:0395:1
          P1:=PIC;                                                                00099500 D        3:0398:1
        END;                                                                      00099600 D        3:0399:1
        IF ICFLG = 1 THEN                                                         00099700 D        3:0399:1
          BEGIN                                                                   00099800 D        3:0400:1
          REPLACE PIC  BY PCN FOR 256;                                            00099900 D        3:0401:1
          WRITE (CRTX,ICODE);                                                     00100000 D        3:0403:1
          ICFLG:= 0;                                                              00100100 D        3:0406:1
        END;                                                                      00100200 D        3:0407:1
      END;                                                                        00100300 D        3:0407:1
      END;                                                                        00100400 D        3:0407:1
    END;                                                                          00100500 D        3:0408:1
SEGMENT    3 IS  409 LONG.
      CLOSE(CKTD,1);                                                              00100600 D        1:0050:2
      CLOSE(TNKD,1);                                                              00100700 D        1:0053:0
      CLOSE(CDIR,1);                                                              00100800 D        1:0055:2
      CLOSE(TDIR,1);                                                              00100900 D        1:0058:0
    END;                                                                          00101000 D        1:0060:2
SEGMENT    1 IS   61 LONG.
SEGMENT    6 IS   44 LONG.
SEGMENT    0 IS    1 LONG.
 0 ERRORS   1018 CARDS.    6061 TOKENS.     11602 RULES.       187 SECONDS.
```

4.4 ESMD Loader

The loader was modified to load ESMD/GAT13-BDS-STACK instead of ESMD/BDS-STACK into GAT13. This change has no effect on the operation of the loader on the TTY output.

```
                $ MERGE ESMD/ESMLDR DISK                                                    00000010          0:0000:*
                %  ESMD LOOP 4 LOADER                                                       00000100 D        0:0000:*
                %                                                                           00000200 D        0:0000:*
                % SOURCE  ESMD/ESMLDR                                                       00000300 D        0:0000:*
                %   OBJECT  ESMLDR                                                         00000400 D        0:0000:*
                $SET ABS   LIST CLIST                                                       00000500 D        0:0000:*
                BEGIN                                                                       00000600 D        0:0000:*
                  FILE CARD(CARDS);                                                         00000700 D        0:0000:*
START OF SEGMENT    1
                  FILE TTY(CRT,POINTER("."),300);                                           00000800 D        1:0004:1
                  NEW NORM,ND,RESP,LPIN,STAT0,STAT1,J,NDPRT,ALL,BAD;                        00000900 D        1:0010:2
                  ARRAY FILENAME[20],INP[20],OUTP[20],LPOUT[46];                            00001000 D        1:0013:0
                  POINTER P1,PFIL,PIN,POUT,PLPOUT,PLPIN,PND;                                00001100 D        1:0015:0
                  DEFINE CRLFETX=4"4D" FOR 1,4"40" FOR 2,4"4A" FOR 1,                       00001200 D        1:0016:3
                  4"43" FOR 1#,                                                             00001300 D        1:0016:3
                    TRUE=1#,                                                                00001400 D        1:0016:3
                  DBUG=BEGIN                                                                00001500 D        1:0016:3
                %  FOR DEBUGGING ONLY ***                                                   00001600 D        1:0016:3
                %   REPLACE POUT BY "STAT0=",STAT0 FOR 3 DIGITS,                            00001700 D        1:0016:3
                %      "STAT1=",STAT1 FOR 3 DIGITS,                                         00001800 D        1:0016:3
                %      "NDPRT=",NDPRT FOR 3 DIGITS,CRLFETX;                                 00001900 D        1:0015:3
                %   WRITE(TTY,POUT);                                                        00002000 D        1:0016:3
                %   FOR DEBUGGING ONLY ***                                                  00002100 D        1:0016:3
                    END#,                                                                   00002200 D        1:0016:3
                    WRLOOP=@26(2,PLPOUT,-46)#,                                              00002300 D        1:0016:3
                  RDLOOP=FOR J:=0 STEP 1 UNTIL VALUE  DO BEGIN END;                         00002400 D        1:0016:3
                   @26(2,PLPIN,1);   J:=FETCH(PLPIN,3);                                     00002500 D        1:0016:3
                    IF J=16 THEN RESP:=0 ELSE RESP:=1#;                                     00002600 D        1:0016:3
                ARRAY PRINT [33], DISKBUFFER [46];                                          00002700 D        1:0016:3
                FILE LINE (PRINTER);                                                        00002800 D        1:0017:3
                FILE NODE (DISK, POINTER ("."),45,180, 182);                                00002900 D        1:0022:2
                FILE MCP (DISK);                                                            00003000 D        1:0030:0
                FILE STACK (DISK, POINTER ("."),45, 180, 40);                               00003100 D        1:0034:3
                POINTER PDSK, STACKNAME, MCPNAME;                                           00003200 D        1:0041:3
                PROCEDURE MIN FORWARD;                                                      00003300 D        1:0042:2
                PROCEDURE SEND (CODE, CODELENGTH); POINTER CODE; NEW CODELENGTH;            00003400 D        1:0042:3
                BEGIN                                                                       00003500 D        1:0045:0
                NEW LENGTH;                                                                 00003600 D        1:0044:3
START OF SEGMENT    3
                POINTER PC;                                                                 00003700 D        3:0000:0
                   LENGTH := MIN (180 - DELTA (POINTER (DISKBUFFER)+4, PDSK), CODELENGTH00003800 D        3:0000:1
                    * 4);                                                                   00003900 D        3:0004:1
                   REPLACE PDSK:PDSK BY PC:CODE FOR LENGTH;                                 00004000 D        3:0005:3
                   IF DELTA (POINTER (DISKBUFFER) + 4, PDSK) = 180 THEN                     00004100 D        3:0009:0
                   BEGIN                                                                    00004200 D        3:0012:1
                    BAD:=0;                                                                 00004300 D        3:0013:1
                    REPLACE PLPOUT BY PND FOR 4;                                            00004400 D        3:0014:1
                    WRLOOP;                                                                 00004500 D        3:0016:1
                    RDLOOP;   %WAIT FOR RESPONSE FR BDS                                     00004600 D        3:0018:0
                    IF RESP=0 THEN BEGIN     %TIMEOUT WITH NO RESPONSE                      00004700 D        3:0031:3
                       REPLACE POUT BY "NO RESPONSE FROM NODE ",                            00004800 D        3:0033:2
                          ND FOR 2 DIGITS, CRLFETX;                                         00004900 D        3:0035:3
                       BAD:=1;                                                              00004950 D        3:0042:0
                       END                                                                  00005000 D        3:0043:0
                    ELSE BEGIN                                                              00005100 D        3:0043:0
                       NDPRT:=FETCH(PLPIN+2,1);                                             00005200 D        3:0044:0
                       STAT0:=FETCH(PLPIN,1);                                               00005300 D        3:0046:1
```

```
                    STAT1:=FETCH(PLPIN+1,1);                                                00005400 D      3:0048:0
                    DBUG;     % FOR DEBUGGING                                               00005500 D      3:0050:1
                    IF ((STAT0.[0:1] OR STAT0.[1:1]) AND                                    00005600 D      3:0050:1
                       (STAT1.[0:1] OR STAT1.[1:1])) THEN                                   00005700 D      3:0053:0
                       BEGIN END     % GOOD TRANSMISSION                                    00005800 D      3:0056:0
                    ELSE BEGIN     % ERROR CONDITION                                        00005900 D      3:0057:0
                       BAD:=1;                                                              00006000 D      3:0058:0
                       IF STAT0.[2:1] THEN REPLACE POUT BY                                  00006100 D      3:0059:0
                            "ERROR-IB0 OVERFLOW IN NODE ",                                  00006200 D      3:0061:3
                           NDPRT FOR 2 DIGITS,CRLFETX                                       00006300 D      3:0063:2
                       ELSE IF STAT0.[3:1] THEN REPLACE POUT BY                             00006400 D      3:0069:0
                            "ERROR-IB1 OVERFLOW IN NODE ",                                  00006500 D      3:0073:2
                           NDPRT FOR 2 DIGITS,CRLFETX                                       00006600 D      3:0075:1
                       ELSE IF STAT0.[4:1] THEN REPLACE POUT BY                             00006700 D      3:0080:3
                            "ERROR-LINE SW PRIM IN NODE ",                                  00006800 D      3:0085:1
                           NDPRT FOR 2 DIGITS,CRLFETX                                       00006900 D      3:0087:0
                       ELSE IF STAT0.[5:1] THEN REPLACE POUT BY                             00007000 D      3:0092:2
                            "ERROR-LINE SW BACK IN NODE ",                                  00007100 D      3:0097:0
                           NDPRT FOR 2 DIGITS,CRLFETX                                       00007200 D      3:0098:3
                       ELSE REPLACE POUT BY                                                 00007300 D      3:0104:1
                            "ERROR-BAD CRC IN NODE ",                                       00007400 D      3:0106:2
                           NDPRT FOR 2 DIGITS,CRLFETX;                                      00007500 D      3:0108:1
                       END;     % FOR BAD STAT                                             00007600 D      3:0114:2
                    END;              % FOR RESP=1                                         00007700 D      3:0114:2
                 IF RESP=0 OR BAD=1 THEN BEGIN                                              00007800 D      3:0114:2
                    WRITE(TTY,POUT);                                                        00007900 D      3:0117:2
                    RETURN (0);                                                             00008000 D      3:0120:2
                 END;                                                                       00008100 D      3:0121:0
               %                                                                            00008200 D      3:0121:0
               %                                                                            00008300 D      3:0121:0
               %                                                                            00008400 D      3:0121:0
               %                                                                            00008500 D      3:0121:0
                    REPLACE PDSK:(PDSK := PCINTER(DISKBUFFER)+4) BY PC FOR CODELENGTH       00008600 D      3:0121:0
                        * 4 - LENGTH;                                                       00008700 D      3:0123:3
                 END;                                                                       00008800 D      3:0126:3
                 SEND := 1;                                                                 00008900 D      3:0126:3
               END;       %SEND                                                            00009000 D      3:0127:3
SEGMENT    3 IS  129 LONG.
               %                                                                            00009100 D      1:0046:0
               %*************************************************************************00009200 D      1:0046:0
               DEFINE                                  % CONSTANTS                          00009300 D      1:0046:0
                   BPW = 4#,                           % BYTES PER WORD                     00009400 D      1:0046:0
                   DATAPACKETLENGTH = 45#,             % STANDARD LENGTH OF DATA PACKETS00009500 D      1:0046:0
                   STACKRECORDSIZE = 45#,                                                   00009600 D      1:0046:0
                   STACKBLOCKSIZE = 180#,                                                   00009700 D      1:0046:0
                   MCPRECORDSIZE = 45#,                                                     00009800 D      1:0046:0
                   MCPBLOCKSIZE = MCPRECORDSIZE#,                                           00009900 D      1:0046:0
                   RECORDINCREMENT = 45#,                                                   00010000 D      1:0046:0
               $OMIT = ABS                                                                  00010100 D      1:0046:0
               $POP OMIT                                                                    00010600 D      1:0046:0
               $OMIT = NOT ABS                                                              00010700 D      1:0046:0
                   EMULATORBASE = 0#,                  % ADDRESS TO START LOADING          00010800 D      1:0046:0
                                                       % STACK MACHINE                    00010900 D      1:0046:0
                   HEADERSIZE = 0#,                                                         00011000 D      1:0046:0
               $POP OMIT                                                                    00011100 D      1:0046:0
                                                                                            00011200 D      1:0046:0
               %   ERROR RETURN CODES                                                       00011300 D      1:0046:0
                                                                                            00011400 D      1:0046:0
                   TRANSMITIONFAILURE =                1#,                                  00011500 D      1:0046:0
                   NONEXISTANTSTACKFILE =              2#,                                  00011600 D      1:0046:0
```

```
                    NONEXISTANTMCPFILE =            3#,                                  00011700 D      1:0046:0
                    OLDSTYLEMCP =                   4#,                                  00011800 D      1:0046:0
                                                                                         00011900 D      1:0046:0
                    DESCRIPTORBASE = [0:16]#;                                            00012000 D      1:0046:0
                                                                                         00012100 D      1:0046:0
                  %*************************************************************************00012200 D      1:0046:0
                  PROCEDURE MIN (X, Y);                                                  00012300 D      1:0046:0
                  %*************************************************************************00012400 D      1:0049:0
                    NEW X, Y;                                                            00012500 D      1:0049:0
                                                                                         00012600 D      1:0049:0
                    MIN := (IF X LSS Y THEN X ELSE Y);                                   00012700 D      1:0049:0
                                                                                         00012800 D      1:0052:3
                  %*************************************************************************00012900 D      1:0052:3
                  PROCEDURE LOADSTACK (STACKFILENAME);                                   00013000 D      1:0052:3
                  %*************************************************************************00013100 D      1:0054:3
                    POINTER                                                              00013200 D      1:0054:3
                      STACKFILENAME;                                                     00013300 D      1:0054:3
                                                % OF EMULATOR TO BE LOADED              00013400 D      1:0054:3
                                                                                         00013500 D      1:0054:3
                  %   VALUE RETURNED = LENGTH OF STACK LOAD IN WORDS IF SUCCESSFULL      00013600 D      1:0054:3
                  %   - ERROR CODE IF FAILURE                                            00013700 D      1:0054:3
                                                                                         00013800 D      1:0054:3
                                                                                         00013900 D      1:0054:3
                    BEGIN                                                                00014000 D      1:0054:3
                    FILE STACK (DISK, STACKFILENAME, STACKRECORDSIZE, STACKBLOCKSIZE);    00014100 D      1:0054:3
START OF SEGMENT      4
                    ARRAY BUFFER [2, HEADERSIZE + STACKRECORDSIZE];                      00014200 D      4:0005:3
                    NEW                                                                  00014300 D      4:0006:3
                      LOADEND,                      % SET WHEN END FLAG DETECTED       00014400 D      4:0006:3
                                                    % OR ERROR CONDINTION              00014500 D      4:0007:0
                      ENDFLAG,                                                           00014600 D      4:0007:0
                      LOADADDRESS := EMULATORBASE,                                       00014700 D      4:0007:1
                      THISRECORD := 0,                                                   00014800 D      4:0007:2
                      NEXTRECORD := 1;                                                   00014900 D      4:0007:3
                                                                                         00015000 D      4:0008:0
                    DEFINE                                                               00015100 D      4:0008:0
                      ADVANCE (RECORD) = RECORD := 1 - RECORD#;                          00015200 D      4:0008:0
                                                                                         00015300 D      4:0008:0
                    IF SEARCH (STACK) LEQ 0 THEN                                         00015400 D      4:0008:0
                      RETURN (- NONEXISTANTSTACKFILE);                                   00015500 D      4:0010:2
                    REPLACE POINTER (ENDFLAG) BY 4"A5" FOR 4;                            00015600 D      4:0012:1
                    READ (STACK, POINTER (BUFFER [NEXTRECORD, HEADERSIZE]));             00015700 D      4:0014:0
                                                                                         00015800 D      4:0017:2
                    DO                                                                   00015900 D      4:0017:2
                      BEGIN                                                              00016000 D      4:0017:2
                      NEW                                                                00016100 D      4:0017:2
START OF SEGMENT      5
                        RECORDLENGTH;                                                    00016200 D      5:0000:*
                                                                                         00016300 D      5:0000:0
                      ADVANCE (THISRECORD);                                              00016400 D      5:0000:0
                      ADVANCE (NEXTRECORD);                                              00016500 D      5:0001:3
                  $OMIT = ABS                                                            00016600 D      5:0003:2
                  $POP OMIT                                                              00016800 D      5:0003:2
                                                                                         00016900 D      5:0003:2
                    IF NOT READ (STACK, POINTER (BUFFER [NEXTRECORD, HEADERSIZE])) THEN  00017000 D      5:0003:2
                      BEGIN                         % LOOK FOR END FLAG                00017100 D      5:0007:3
                      WHILE RECORDLENGTH LSS STACKRECORDSIZE * BPW AND                   00017200 D      5:0008:3
                          POINTER (BUFFER [THISRECORD, HEADERSIZE]) +                    00017300 D      5:0010:0
                            RECORDLENGTH NEQ POINTER (ENDFLAG) FOR 4 DO                  00017400 D      5:0011:3
                        RECORDLENGTH := RECORLLENGTH + 1;                                00017500 D      5:0014:0
```

```
                    RECORDLENGTH := (RECORDIENGTH - 1) DIV BPW + 1;                    00017600 D       5:0016:2
                                                  % CONVERT BYTE COUNT TO NEXT         00017700 D       5:0018:1
                                                  % EVEN WORD COUNT                    00017800 D       5:0018:1
                    LOADEND := TRUE;                                                  00017900 D       5:0018:1
                    END                                                               00018000 D       5:0019:1
                  ELSE                                                                00018100 D       5:0019:1
                    RECORDLENGTH := STACKRECORDSIZE;                                  00018200 D       5:0020:1
                  IF NOT SENI (POINTER (BUFFER [THISRECORD, 0]), RECORDLENGTH) THEN   00018300 D       5:0021:0
                      BEGIN                                                           00018400 D       5:0024:0
                      LOADSTACK := - TRANSMITIONFAILURE;                              00018500 D       5:0025:0
                      LOADENI := TRUE;                                                00018600 D       5:0026:1
                      END                                                             00018700 D       5:0027:1
                  ELSE                                                                00018800 D       5:0027:1
                      BEGIN                                                           00018900 D       5:0028:1
                      LOADADDRESS := LOADADDRESS + RECORDLENGTH;                      00019000 D       5:0028:1
                      IF LOADEND THEN                                                 00019100 D       5:0030:0
                        LOADSTACK := LOADADIRESS;                                     00019200 D       5:0030:2
                      END;                                                            00019300 D       5:0032:3
                                                                                      00019400 D       5:0032:3
                  END UNTIL LOADEND;                                                  00019500 D       5:0032:3
SEGMENT    5 IS   34 LONG.
                  END;                                                                00019600 D       4:0019:1
SEGMENT    4 IS   20 LONG.
                                                                                      00019700 D       1:0056:0
              $OMIT = NOT ABS                                                         00019800 D       1:0056:0
              %*************************************************************************00019900 D       1:0056:0
              PROCEDURE SENDZEROS (NUMBEROFZEROS);                                    00020000 D       1:0056:0
              %*************************************************************************00020100 D       1:0058:0
                NEW NUMBEROFZEROS;                                                    00020200 D       1:0058:0
                                                                                      00020300 D       1:0058:0
                  BEGIN                                                               00020400 D       1:0058:0
                  ARRAY BUFFER [DATAPACKETLENGTH];                                    00020500 D       1:0058:0
START OF SEGMENT   6
                                                                                      00020600 D       6:0000:1
                  WHILE NUMBEROFZEROS GTR 0 DO                                        00020700 D       6:0000:1
                    BEGIN                                                             00020800 D       6:0001:1
                    IF NOT SEND (POINTER (BUFFER) , MIN (DATAPACKETLENGTH,            00020900 D       6:0002:1
                        NUMBEROFZEROS)) THEN                                          00021000 D       6:0004:1
                      RETURN (FALSE);                                                 00021100 D       6:0005:2
                    NUMBEROFZEROS := NUMBEROFZEROS - DATAPACKETLENGTH;                00021200 D       6:0007:0
                    END;                                                              00021300 D       6:0008:3
                  RETURN (TRUE);                                                      00021400 D       6:0009:1
                  END;                                                                00021500 D       6:0009:3
SEGMENT    6 IS   11 LONG.
              $POP OMIT                                                               00021600 D       1:0059:1
                                                                                      00021700 D       1:0059:1
              %*************************************************************************00021800 D       1:0059:1
              PROCEDURE MCPLOAD (MCPFILENAME, MEMORYSIZE, STACKBASE);                 00021900 D       1:0059:1
              %*************************************************************************00022000 D       1:0061:3
                POINTER                                                               00022100 D       1:0061:3
                  MCPFILENAME;                    % POINTER TO STRING FOR FILE NAME00022200 D       1:0061:3
                                                  % OF MCP TO BE LOADED. 0 IF NONE 00022300 D       1:0061:3
                NEW                                                                   00022400 D       1:0061:3
                  MEMORYSIZE,                     % TOTAL MEMORY SIZE               00022500 D       1:0061:3
                  STACKBASE;                      % ADDRESS TO START RUN TIME STACK00022600 D       1:0061:3
                                                                                      00022700 D       1:0061:3
                BEGIN                                                                 00022800 D       1:0061:3
                FILE MCP (DISK, MCPFILENAME, MCPRECORDSIZE, MCPBLOCKSIZE);            00022900 D       1:0061:1
START OF SEGMENT   7
                ARRAY BUFFER [HEADERSIZE + MCPRECORDSIZE];                            00023000 D       7:0005:1
```

```
                NEW                                                                    00023100 D        7:0005:3
                  RELOCATIONOFFSET,                                                    00023200 D        7:0005:3
                  SEGMENTDICTIONARYLENGTH,                                             00023300 D        7:0006:0
                  CODESEGMENTLENGTH,                                                   00023400 D        7:0006:1
                  LOADADDRESS;                                                         00023500 D        7:0006:2
                                                                                       00023600 D        7:0006:3
                IF SEARCH (MCP) LEQ 0 THEN                                             00023700 D        7:0006:3
                  RETURN (- NONEXISTANTMCPFILE);                                       00023800 D        7:0009:1
                READ (MCP, BUFFER);                                                    00023900 D        7:0011:0
                IF (SEGMENTDICTIONARYLENGTH := - BUFFER [0]) LSS 0 THEN                00024000 D        7:0013:3
                  RETURN (- OLDSTYLEMCP);                                              00024100 D        7:0015:2
                CODESEGMENTLENGTH := BUFFER [1];                                       00024200 D        7:0017:1
                                                                                       00024300 D        7:0018:1
                READ (MCP, POINTER (BUFFER [HEADERSIZE]));                             00024400 D        7:0018:1
                RELOCATIONOFFSET := MEMORYSIZE - CODESEGMENTLENGTH;                    00024500 D        7:0021:1
                                                                                       00024600 D        7:0022:3
                BEGIN NEW INDEX;FOR INDEX := 0 STEP 1 UNTIL SEGMENTDICTIONARYLENGTH -100024700 D        7:0022:3
SEGMENT    2 IS   64 LONG.
START OF SEGMENT    8
                  DO                                                                   00024800 D        8:0003:0
                  BUFFER [HEADERSIZE + INDEX] := * &                                   00024900 D        8:0004:2
                    (BUFFER [HEADERSIZE + INDEX].DESCRIPTORBASE +                      00025000 D        8:0006:2
                      RELOCATIONOFFSET) DESCRIPTORBASE;                                00025100 D        8:0008:2
                  END;                                                                 00025200 D        8:0010:3
SEGMENT    8 IS   12 LONG.
                                                                                       00025300 D        7:0023:3
              $OMIT = ABS                                                              00025400 D        7:0023:3
              $POP OMIT                                                                00025600 D        7:0023:3
                                                                                       00025700 D        7:0023:3
                IF NOT SEND (POINTER (BUFFER), SEGMENTDICTIONARYLENGTH) THEN           00025800 D        7:0023:3
                  RETURN (- TRANSMITIONFAILURE);                                       00025900 D        7:0025:3
                                                                                       00026000 D        7:0027:2
              $OMIT = NOT ABS                                                          00026100 D        7:0027:2
                                                  % SEND ZEROS BETWEEN SEGMENT         00026200 D        7:0027:2
                                                  % SEGMENT DICITONARY AND CODE        00026300 D        7:0027:2
                                                  % SEGMENTS                           00026400 D        7:0027:2
                IF NOT SENDZEROS (MEMORYSIZE -                                         00026500 D        7:0027:2
                    (STACKBASE + SEGMENTDICTIONARYLENGTH + CODESEGMENTLENGTH)) THEN    00026600 D        7:0028:2
                  RETURN (- TRANSMITIONFAILURE);                                       00026700 D        7:0030:3
              $POP OMIT                                                                00026800 D        7:0032:2
                                                                                       00026900 D        7:0032:2
                LOADADDRESS := RELOCATIONOFFSET;     % SEND CODE SEGMENTS             00027000 D        7:0032:2
                WHILE READ (MCP, POINTER (BUFFER [HEADERSIZE])) AND                    00027100 D        7:0033:1
                    CODESEGMENTLENGTH GTR 0 DO                                         00027200 D        7:0036:0
                  BEGIN                                                                00027300 D        7:0037:0
              $OMIT = ABS                                                              00027400 D        7:0038:0
              $POP OMIT                                                                00027600 D        7:0038:0
                                                                                       00027700 D        7:0038:0
                  IF NOT SEND (POINTER (BUFFER), MIN (MCPRECORDSIZE,                   00027800 D        7:0038:0
                     CODESEGMENTLENGTH)) THEN                                          00027900 D        7:0040:0
                    RETURN (- TRANSMITIONFAILURE);                                     00028000 D        7:0041:0
                                                                                       00028100 D        7:0042:3
              $OMIT = ABS                                                              00028200 D        7:0042:3
              $POP OMIT                                                                00028400 D        7:0042:3
                                                                                       00028500 D        7:0042:3
                  CODESEGMENTLENGTH := CODESEGMENTLENGTH - MCPRECORDSIZE;              00028600 D        7:0042:3
                  END;                                                                 00028700 D        7:0044:0
                                                                                       00028800 D        7:0044:2
                RETURN (TRUE);                                                         00028900 D        7:0044:2
                END;                                                                   00029000 D        7:0045:0
```

```
SEGMENT    7 IS   46 LONG.
                                                                                00029100 D        1:0062:2
                %**************************************************************00029200 D        1:0062:2
                PROCEDURE STACKLOAD (STACKFILENAME, MCPFILENAME, MEMORYSIZE);    00029300 D        1:0062:2
                %**************************************************************00029400 D        1:0065:0
                  POINTER                                                       00029500 D        1:0065:0
                    STACKFILENAME,                  % POINTER TO STRING FOR FILE NAME00029600 D  1:0065:0
                                                    % OF EMULATOR TO BE LOADED   00029700 D        1:0065:0
                    MCPFILENAME;                    % POINTER TO STRING FOR FILE NAME00029800 D  1:0065:0
                                                    % OF MCP TO BE LOADED. 0 IF NONE 00029900 D  1:0065:0
                  NEW                                                           00030000 D        1:0065:0
                    MEMORYSIZE;                     % TOTAL MEMORY SIZE         00030100 D        1:0065:0
                                                                                00030200 D        1:0065:0
                  BEGIN                                                         00030300 D        1:0065:0
                  NEW                                                           00030400 D        1:0064:2
START OF SEGMENT   10
                    RESULT,                                                     00030500 D       10:0000:*
                    STACKBASE;                      % ADDRESS TO START RUN TIME STACK00030600 D 10:0000:0
                                                                                00030700 D       10:0000:1
                  IF (RESULT := STACKBASE := LOADSTACK (STACKFILENAME)) GEQ 0 THEN 00030800 D    10:0000:1
                    BEGIN                                                       00030900 D       10:0003:0
                    IF ABS (MCPFILENAME) NEQ 0 THEN     % IS THERE AN MCP TO BE LOADED 00031000 D 10:0004:0
                      RESULT := MCPLOAD (MCPFILENAME, MEMORYSIZE + EMULATORBASE, 00031100 D      10:0005:1
                        STACKBASE);                                             00031200 D       10:0008:2
                $OMIT = ABS                                                     00031300 D       10:0009:1
                $POP OMIT                                                       00032500 D       10:0009:1
                    END                                                         00032600 D       10:0009:1
                  ELSE                                                          00032700 D       10:0009:1
                    STACKLOAD := RESULT;            % ERROR CONDINTION         00032800 D       10:0010:1
                  END;                                                          00032900 D       10:0011:1
SEGMENT   10 IS   12 LONG.
                %                                                               00033000 D        1:0065:3
                %                                                               00033100 D        1:0065:3
                 IF VALUE=-1 THEN VALUE:=100;                                   00033200 D        1:0065:3
                PLPIN:=POINTER(LPIN); % LOOP INPUT PTR                          00033300 D        1:0069:2
                PND:=POINTER(ND);                                               00033400 D        1:0070:3
                PLPOUT := POINTER (DISKBUFFER);  %OUTPUT BUFFER                 00033500 D        1:0072:0
                PFIL:=POINTER(FILENAME);                                        00033600 D        1:0074:0
                PIN:=POINTER(INP);                                              00033700 D        1:0075:3
                POUT:=POINTER(OUTP);                                            00033800 D        1:0077:2
                ALL:=0;                                                         00033900 D        1:0079:1
                REPLACE POUT BY "IS THIS A NORMAL LOAD?",CRLFETX;               00034000 D        1:0080:0
                WRITE(TTY,POUT);                                                00034100 D        1:0087:2
                READ(CARD,PIN);                                                 00034200 D        1:0090:1
                IF PIN="YES" THEN                                               00034300 D        1:0093:0
                  BEGIN                                                         00034400 D        1:0095:0
                  NORM:=1;                                                      00034500 D        1:0096:0
                  ND:=12;                                                       00034600 D        1:0096:3
                  REPLACE PFIL BY "GAT12.";                                     00034700 D        1:0097:2
                  END                                                           00034800 D        1:0100:0
                  ELSE BEGIN                                                    00034900 D        1:0100:0
                  NORM:=0;                                                      00035000 D        1:0101:0
                  REPLACE POUT BY "ENTER ND OF NODE TO BE LOADED",CRLFETX;      00035100 D        1:0101:3
                  WRITE(TTY,POUT);                                              00035200 D        1:0109:1
                  READ(CARD,PIN);                                               00035300 D        1:0112:0
                  IF PIN="ALL" THEN BEGIN ALL:=1; ND:=12; END                   00035400 D        1:0114:3
                  ELSE BEGIN                                                    00035500 D        1:0119:1
                  ND:=INT(PIN,2);                                               00035600 D        1:0120:1
                    IF ND=16 THEN VALUE:=25;                                    00035700 D        1:0121:3
                  IF ND LEQ 11 OR ND GEQ 20 THEN                                00035800 D        1:0124:2
```

```
                    BEGIN                                                                   00035900 D        1:0126:1
                    REPLACE POUT BY "INVALID ND-RERUN ESMLDR",CRLFETX;                      00036000 D        1:0127:1
                $LIST                                                                       00036010          1:0134:3
                    WRITE(TTY,POUT);                                                        00036100 D        1:0134:3
                    RETURN;   %STOP                                                         00036200 D        1:0137:2
                    END;                                                                    00036300 D        1:0137:3
                  END;                                                                      00036400 D        1:0137:3
                    REPLACE POUT BY "ENTER OBJ FILE TO BE LOADED",CRLFETX;                  00036500 D        1:0137:3
                    WRITE(TTY,POUT);                                                        00036600 D        1:0145:1
                    READ(CARD,PFIL);                                                        00036700 D        1:0148:0
                    SCAN P1:PFIL UNTIL " ";                                                 00036800 D        1:0150:3
                    REPLACE P1 BY ".";                                                      00036900 D        1:0153:2
                    END;                                                                    00037000 D        1:0155:0
                WHILE TRUE DO                                                               00037100 D        1:0155:0
                BEGIN                                                                       00037200 D        1:0155:1
                   CLOSE(STACK,1);                                                          00037300 D        1:0156:1
                   IF PFIL="SDL14." OR PFIL="SDL15." THEN                                   00037400 D        1:0159:0
                   FILEATTRIBUTE(STACK,1,POINTER("ESMD/SDLC-BDS-STACK."))                   00037500 D        1:0163:1
                   ELSE IF PFIL="SEC19." THEN                                               00037600 D        1:0167:2
                   FILEATTRIBUTE(STACK,1,POINTER("ESMD/PDP-BDS-STACK."))                    00037700 D        1:0171:0
                   ELSE IF PFIL="GAT13." THEN                                               00037720          1:0175:1
                   FILEATTRIBUTE(STACK,1,POINTER("ESMD/GAT13-BDS-STACK."))                  00037730          1:0178:3
                   ELSE                          % NORMAL STACK MACHINE                     00037800 D        1:0183:0
                   FILEATTRIBUTE(STACK,1,POINTER("ESMD/BDS-STACK."));                       00037900 D        1:0184:2
SEGMENT    9 IS   66 LONG.
                   STACKNAME := (PDSK := POINTER (DISKBUFFER) + 4);                         00038000 D        1:0188:1
                $ LIST                                                                      00038010          1:0191:3
                   REPLACE STACKNAME BY STACK,".";                                          00038100 D        1:0191:3
                  IF STACKLOAD (STACKNAME, PFIL, 8190) GEQ 0 AND BAD=0 THEN                 00038200 D        1:0195:3
                   BEGIN                                                                    00038300 D        1:0199:2
                     REPLACE POUT BY "SUCCESSFUL LOAD-NODE ",ND FOR 2 DIGITS,CRLFETX;       00038400 D        1:0200:2
                      WRITE (TTY, POUT);                                                    00038500 D        1:0208:3
                   END;                                                                     00038600 D        1:0211:2
                   IF NORM = 0 AND ALL = 0 THEN RETURN;   %CHECK FOR END OF PROGRAM         00038700 D        1:0211:2
                   ND := ND + 1;                                                            00038800 D        1:0214:0
                   IF ND=16 THEN ND:=17;                                                    00038900 D        1:0215:0
                    IF ND=20 THEN ND:=16;                                                   00039000 D        1:0217:2
                  IF ALL=1 THEN BEGIN END                                                   00039100 D        1:0220:0
                    ELSE BEGIN                                                              00039200 D        1:0221:3
                    IF ND=13 THEN REPLACE PFIL BY "GAT13."                                  00039300 D        1:0222:3
                    ELSE IF ND=14 THEN REPLACE PFIL BY "SDL14."                             00039400 D        1:0225:0
                    ELSE IF ND=15 THEN REPLACE PFIL BY "SDL15."                             00039500 D        1:0230:1
                    ELSE IF ND=17 THEN REPLACE PFIL BY "GAT17."                             00039600 D        1:0235:2
                    ELSE IF ND=18 THEN REPLACE PFIL BY "CRT18."                             00039700 D        1:0240:3
                    ELSE IF ND=19 THEN REPLACE PFIL BY "SEC19."                             00039800 D        1:0246:0
                    ELSE IF ND=16 THEN REPLACE PFIL BY "HST16.";                            00039900 D        1:0251:1
                   END;                                                                     00040000 D        1:0258:2
                   IF ND=16 THEN BEGIN NORM:=0; ALL:=0; END;                                00040100 D        1:0258:2
                END;                                                                        00040200 D        1:0261:3
                END ?                                                                       00040300 D        1:0262:3
SEGMENT    1 IS  264 LONG.
SEGMENT   11 IS   24 LONG.
SEGMENT    0 IS    1 LONG.
 0 ERRORS    390 CARDS.   1919 TOKENS.      3579 RULES.       100 SECONDS.
 PROGRAM SIZE =    683 WORDS   18 DISK SECTORS.
```

APPENDIX 1 Systems Drawings

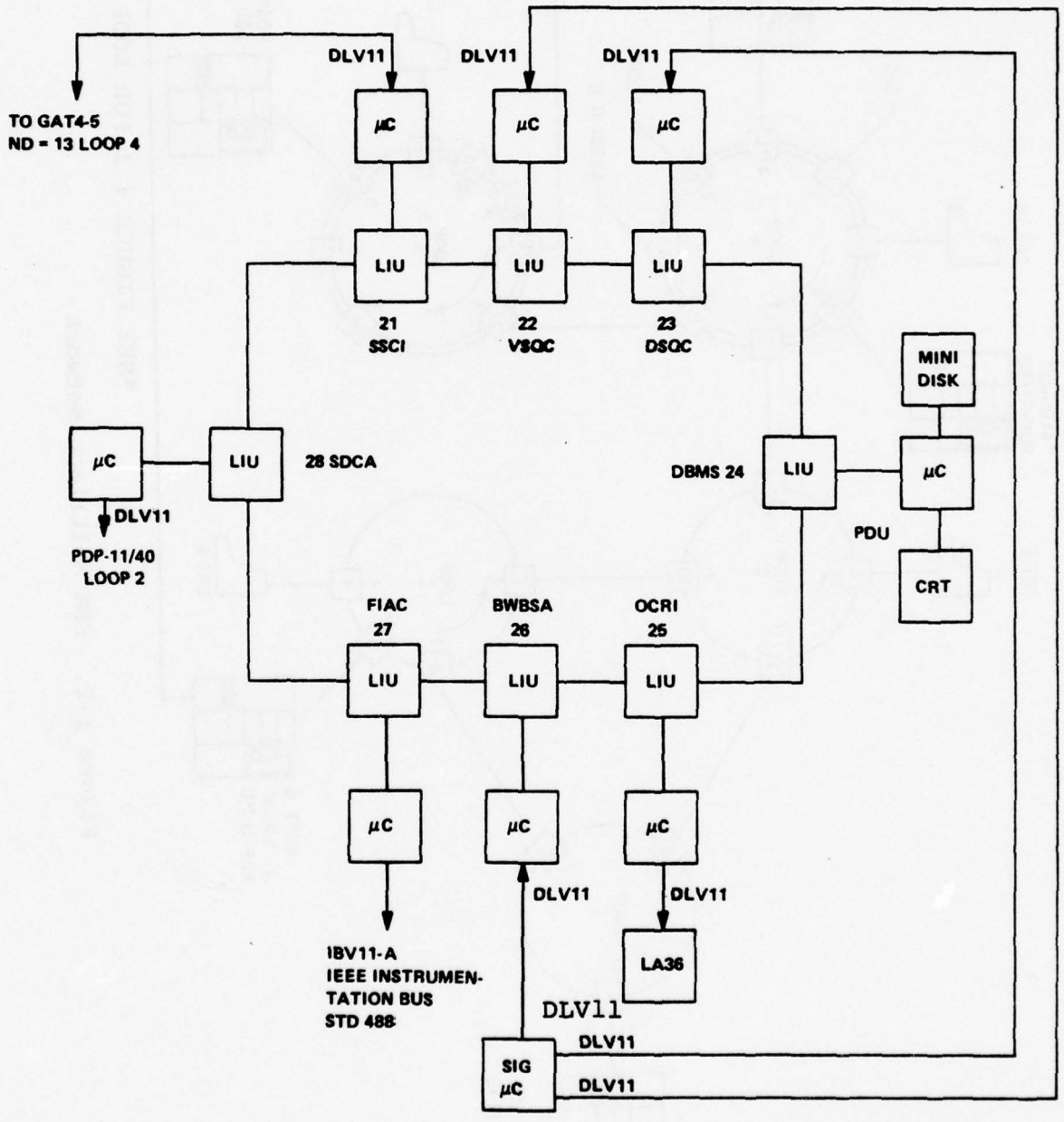
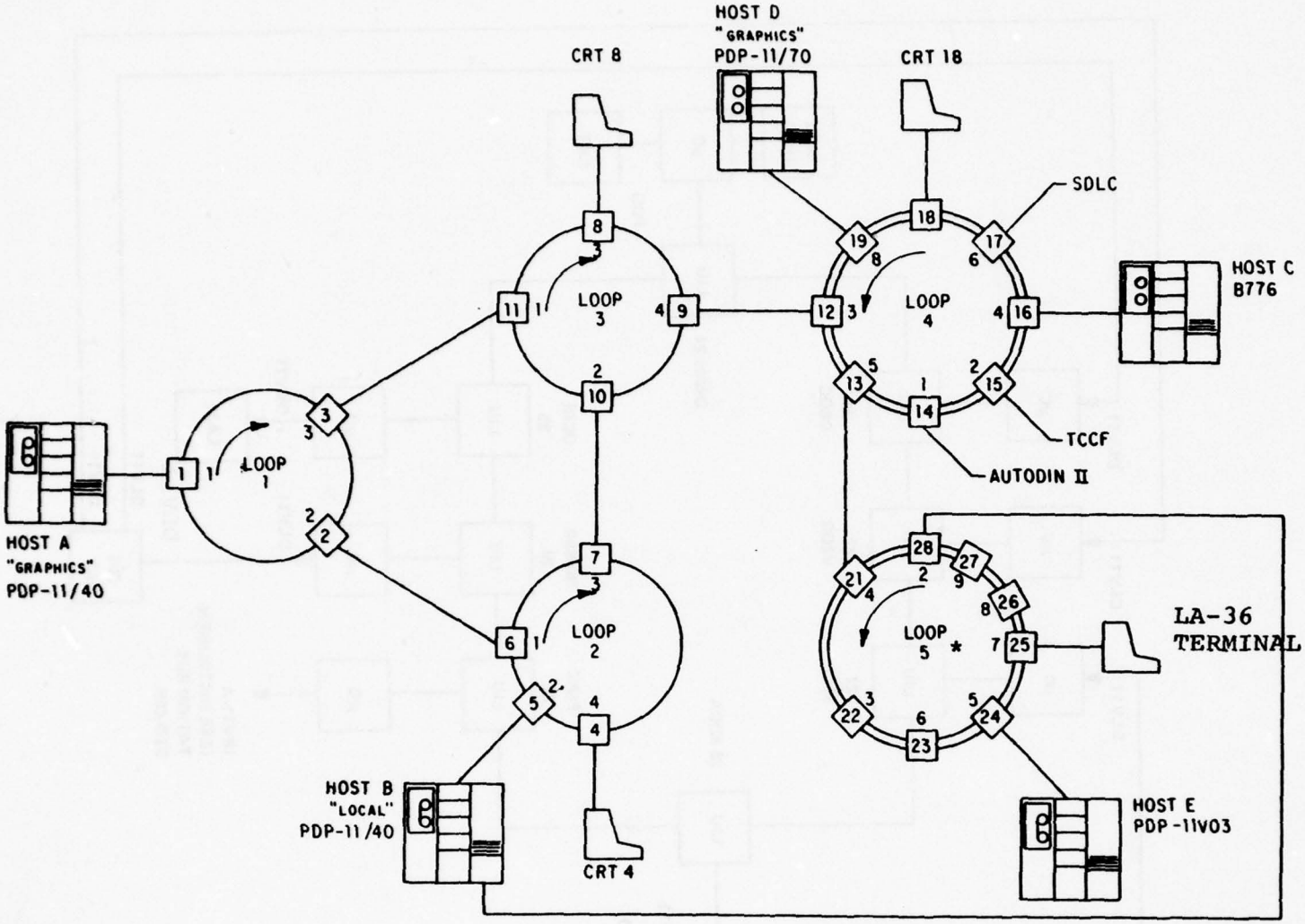


Figure 1-1. MSCDM System Configuration



*SEE FIGURE 1.1 FOR LOOP CONFIGURATION

Figure 1-2. ESM Multiloop Network

APPENDIX 2 System Diskettes

System Diskettes

The MSCDM system diskettes are described below.

Diskette #1: Contains the Loop Loader Utility and the RT-11 operating system. This diskette normally resides in drive 0 (DX0:).

Diskette #2: Contains the task files for Nodes 21 - 27 (excluding Node 24) and for the SIG

Diskette #2A: Contains the task files for Node 28.

Diskette #3: Contains DBMS run file and S/J OS and nodal task file.

Diskette #3A: Contains the message display file, and status file for Node 24 execution.

Diskette #4: Used for linking. Contains FORTRAN compiler, MACRO Assembler and libraries. Runs in DX0:

Diskette #5: Contains the source files for Node 21.

Diskette #6: Contains the source files for Node 22.

Diskette #7: Contains the source files for Node 23.

Diskette #8: Contains the source files for Node 24.

Diskette #9: Contains the source files for Node 25.

Diskette #10: Contains the source files for Node 26.

Diskette #11: Contains the source files for Node 27.

Diskette #12: Contains the source files for Node 28.

Diskette #13: Contains the source files for the SIG.

Diskette #14: Scratch disk.

Diskette #15: Contains the source files for the Loader Utility and PROM Loader programs.

Diskette #16: Contains diagnostic programs for the LSI-11 processor and loop hardware.

Diskette #17: Contains miscellaneous files, IEEE interface and other special programs.

Diskette #18: Contains DBMS Linker DX0:

Diskette #18A: Contains DBMS Object files.

```
decwriter program:     DISK1.DIR     23-Jul-79     11:33:25     Page 001


23-Jul-79
DXMNFB.SYS      97 18-Oct-78      TT    .SYS       2 18-Oct-78
DX    .SYS       2 18-Oct-78      NL    .SYS       2 18-Oct-78
MACRO .SAV      45 18-Oct-78      STARTF.COM       1
PIP   .SAV      16 18-Oct-78      DIR   .SAV      17 18-Oct-78
DUP   .SAV      17 18-Oct-78      PGLOOP.SAV      30
EDIT  .SAV      21 18-Oct-78      LINK  .SAV      29 18-Oct-78
DUMP  .SAV       7 18-Oct-78      FORTRA.SAV     128 27-Dec-78
FDMLDR.SAV      32 27-Apr-79      DUMMY .MAC       1
 16 Files, 447 Blocks
 33 Free Blocks
```

```
23-Jul-79
SIGGEN.LDA      36 22-May-79      NODE21.SAV      57 22-May-79
NODE22.SAV      74 22-May-79      NODE23.SAV      73 22-May-79
NODE25.SAV      65 22-May-79      NODE26-SAV      72 22-May-79
NODE27.SAV      75 22-May-79
 7 Files, 452 Blocks
28 Free Blocks
```

```
decwriter program:     DISK2A.DIR     23-Jul-79     11:36:51     Page 001


23-Jul-79
NODE28.SAV      72 22-May-79
 1 Files, 72 Blocks
 408 Free Blocks
```

```
23-Jul-79
DXMNSJ.SYS      86 17-May-79      TT     .SYS       2 17-May-79
DX     .SYS      2 17-May-79      NL     .SYS       2 17-May-79
PIP    .SAV     16 17-May-79      DUP    .SAV      17 17-May-79
DIR    .SAV     17 17-May-79      STARTS.COM        1 17-May-79
NODE24.SAV      93 17-May-79
 9 Files, 236 Blocks
 244 Free Blocks
```

```
decwriter program:     DISK3A.DIR     23-Jul-79     11:32:13     Page 001


23-Jul-79
BDSTAT.FOR       5 17-May-79      STAT  .COM       1 17-May-79
BDSTAT.SAV      20 17-May-79      STATUS.MAS      40 17-May-79
MASDIR.DAT       1 17-May-79      MSGCON.FOR       3 17-May-79
MSGCON.SAV      19 17-May-79      MSGL  .DAT      21 17-May-79
MSG   .DAT      51 17-May-79      CRASH .COM       1 14-May-79
BACKUP.COM       1 14-May-79      DIR   .DAT       9 17-May-79
DAT   .DAT      32 17-May-79      CKTD  .DAT      32 17-May-79
REPORT.DAT       9 17-May-79      TDIR  .DAT       9 17-May-79
MASDAT.BAK       1 17-May-79      MASDAT.DAT       2 17-May-79
TNKD  .DAT      32 17-May-79      CDIR  .DAT       9 17-May-79
TDIR  .OLD       9 17-May-79      TNKD  .OLD      32 17-May-79
CDIR  .OLD       9 17-May-79      CKTD  .OLD      32 17-May-79
STATUS.DAT      40 17-May-79
 25 Files, 420 Blocks
 60 Free Blocks
```

```
decwriter program:     DISK4.DIR     01-Jul-79     12:00:00     Page 001


01-Jul-79
DXMNFB.SYS      97 08-Feb-79      TT     .SYS       2 21-Feb-79
DX     .SYS      2 21-Feb-79      NL     .SYS       2 21-Feb-79
STARTF.COM       1 21-Feb-79      PIP    .SAV      16 21-Feb-79
DIR    .SAV     17 21-Feb-79      DUMP   .SAV       7 18-Oct-78
LINK   .SAV     29 21-Feb-79      SYSLIB.OBJ     203 27-Dec-78
 10 Files, 376 Blocks
 104 Free Blocks
```

```
decwriter program:     DISK5.DIR     01-Jul-79     12:00:00     Page 001


01-Jul-79
NODAL1.OLD      10 23-Jul-79     NODAL2.OLD      17 25-Apr-79
NODAL .OLD       6 23-Jul-79     FDM    .OLD     41 25-Apr-79
NODAL .FOR       8 22-May-79     NODAL1.FOR      10 22-May-79
FDM    .MAC     41 21-May-79     COMP21.COM       1 23-Jul-79
LNK21 .COM       1 23-Jul-79
 9 Files, 135 Blocks
 345 Free Blocks
```

```
decwriter program:     DISK6.DIR     01-Jul-79     12:00:00     Page 001


01-Jul-79
NODAL .OLD       7 04-Apr-79     NODAL1.OLD      10 04-Apr-79
VSQC  .OLD      15 16-Apr-79     FDM   .OLD      41 15-Apr-79
NODAL2.OLD      25 01-May-79     NODAL .FOR       8 22-May-79
NODAL1.FOR      17 22-May-79     VSQC  .FOR      13 22-May-79
FDM   .MAC      41 21-May-79     COMP22.COM       1 01-Jul-79
LNK22 .COM       1 01-Jul-79
 11 Files, 179 Blocks
 301 Free Blocks
```

```
decwriter program:     DISK7.DIR     01-Jul-79     12:00:00     Page 001


01-Jul-79
NODAL .OLD       7 04-Apr-79     NODAL1.OLD      10 04-Apr-79
FDM   .OLD      41 12-Apr-79     NODAL2.OLD      25 01-May-79
DSQC  .OLD      13 14-May-79     NODAL .FOR       9 21-May-79
NODAL1.FOR      17 22-May-79     DSQC  .FOR      12 22-May-79
FDM   .MAC      41 21-May-79     COMP23.COM       1 01-Jul-79
LNK23 .COM       1 01-Jul-79
 11 Files, 177 Blocks
 303 Free Blocks
```

01-Jul-79

M2000 .OLD	6	22-May-79	M4000 .OLD	22	22-May-79
FDM .OLD	40	22-May-79	NODAL .OLD	9	22-May-79
NODAL1.OLD	18	22-May-79	M0000 .OLD	19	22-May-79
M1000 .OLD	6	22-May-79	M3000 .OLD	6	22-May-79
M5000 .OLD	21	22-May-79	M6000 .OLD	8	22-May-79
COMP2 .COM	1	17-May-79	COMP2A.COM	1	17-May-79
COMP1 .COM	1	17-May-79	M2000 .FOR	6	17-May-79
M4000 .FOR	22	17-May-79	NODAL .FOR	9	17-May-79
NODAL1.FOR	18	17-May-79	M0000 .FOR	19	17-May-79
M1000 .FOR	6	17-May-79	M3000 .FOR	6	17-May-79
M5000 .FOR	21	17-May-79	M6000 .FOR	8	17-May-79
FDM .MAC	40	17-May-79			

23 Files, 313 Blocks
167 Free Blocks

```
decwriter program:     DISK9.DIR     01-Jul-79     12:00:00     Page 001


01-Jul-79
NODAL1.OLD      13 23-Jul-79      NODAL .OLD       7 08-May-79
FDM    .OLD     42 09-May-79      NODAL2.OLD      24 09-May-79
NODAL .FOR      11 21-May-79      NODAL1.FOR      15 22-May-79
FDM    .MAC     42 21-May-79      COMP25.COM       1 01-Jul-79
LNK25 .COM       1 01-Jul-79
 9 Files, 156 Blocks
 324 Free Blocks
```

```
01-Jul-79
NODAL1.OLD      10 04-Apr-79      NODAL .OLD       6 04-Apr-79
FDM    .OLD     41 23-Jul-79      NODAL2.OLD      25 01-May-79
BWBSA .OLD      13 01-May-79      NODAL .FOR       9 21-May-79
NODAL1.FOR      17 22-May-79      BWBSA .FOR      13 22-May-79
FDM    .MAC     41 21-May-79      COMP26.COM       1 01-Jul-79
LNK26 .COM       1 01-Jul-79
 11 Files, 177 Blocks
 303 Free Blocks
```

```
01-Jul-79
NODAL1.OLD      10 04-Apr-79     NODAL .OLD       6 25-Apr-79
NODAL2.OLD      25 23-Jul-79     FIAC  .OLD      19 06-May-79
FDM   .OLD      41 01-May-79     NODAL .FOR       9 21-May-79
NODAL1.FOR      17 21-May-79     FIAC  .FOR      17 22-May-79
FDM   .MAC      41 21-May-79     COMP27.COM       1 01-Jul-79
LNK27 .COM       1 01-Jul-79
 11 Files, 187 Blocks
 293 Free Blocks
```

```
01-Jul-79
NODAL1.OLD      10 04-Apr-79      FDM    .OLD      41 23-Jul-79
NODAL .OLD       7 23-Jul-79      NODAL2.OLD       26 23-Jul-79
SDCA  .OLD      13 06-May-79      NODAL .FOR        9 21-May-79
NODAL1.FOR      17 21-May-79      SDCA  .FOR       14 21-May-79
FDM   .MAC      41 21-May-79      COMP28.COM        1 01-Jul-79
LNK28 .COM       1 01-Jul-79
 11 Files, 180 Blocks
 300 Free Blocks
```

```
decwriter program:     DISK13.DIR     01-Jul-79     12:00:00     Page 001


01-Jul-79
LNK20 .COM       1 07-Mar-79     SIGMAC.OLD       3 12-Apr-79
SIGGEN.OLD      17 01-May-79     SIGGEN.CRT      44 01-May-79
SIGGEN.LDA      36 14-May-79     SIGGEN.FOR      17 21-May-79
SIGMAC.MAC       3 21-May-79     COMP20.COM       1 01-Jul-79
 8 Files, 122 Blocks
 358 Free Blocks
```

```
01-Jul-79
LDRMAC.OLD      17 01-Feb-79     PROM26.MAC      19 01-Feb-79
PROM21.MAC      17 01-Feb-79     PROM22.MAC      17 01-Feb-79
PROM23.MAC      17 01-Feb-79     PROM27.MAC      17 01-Feb-79
PROM28.MAC      17 01-Feb-79     FDMLDR.COM       1 06-Feb-79
PROM25.MAC      18 08-Feb-79     SIGLDR.MAC       7 11-Feb-79
DUMMY .MAC       0 12-Feb-79     FDMLDR.OLD      16 27-Apr-79
FDMLDR.FOR      16 22-Apr-79     LDRMAC.MAC      17 22-May-79
FDMLDR.OBJ      52 27-Apr-79     LDRMAC.OBJ       3 27-Apr-79
FDMLDR.LST      36 01-Jul-79
 17 Files, 287 Blocks
 193 Free Blocks
```

```
01-Jul-79
PRTCT.MAC       15 01-Feb-79     LSICLK.MAC       4 01-Feb-79
CRTPRT.MAC      15 01-Feb-79     LSIMEM.MAC       8 06-Feb-79
LSICPU.MAC      11 06-Feb-79     LIUINT.MAC      10 06-Feb-79
LIUBUF.MAC      12 06-Feb-79     LIURAM.MAC      12 06-Feb-79
DIARCV.MAC      20 17-Mar-79     DIASND.MAC      10 17-Mar-79
EXEML .COM       1 17-Mar-79     LSICPU.OBJ       2 17-Mar-79
LSIMEM.OBJ       1 17-Mar-79     LSICLK.OBJ       1 17-Mar-79
LIUINT.OBJ       2 17-Mar-79     LIUBUF.OBJ       2 17-Mar-79
LIURAM.OBJ       2 17-Mar-79     DIARCV.OBJ       3 17-Mar-79
DIASND.OBJ       2 17-Mar-79     CRTPRT.OBJ       3 17-Mar-79
PRTCRT.OBJ       3 17-Mar-79     LSICPU.SAV       2 17-Mar-79
LSIMEM.SAV       2 17-Mar-79     LSICLK.SAV       2 17-Mar-79
LSICPU.LDA       1 17-Mar-79     LSIMEM.LDA       1 17-Mar-79
LSICLK.LDA       1 17-Mar-79     LIURAM.SAV       3 17-Mar-79
LIUBUF.SAV       4 17-Mar-79     LIUINT.SAV       3 17-Mar-79
CRTPRT.SAV       5 17-Mar-79     PRTCRT.SAV       5 17-Mar-79
DIARCV.SAV       4 17-Mar-79     DIASND.SAV       3 17-Mar-79
DIR   .DIR       3 17-Mar-79
 35 Files, 178 Blocks
 302 Free Blocks
```

```
01-Jul-79
SIEEE .MAC      10 03-Feb-79     RIEFE .MAC      15 03-Feb-79
PGMAC .MAC      17 12-Feb-79     PGLOOP.COM       1 12-Feb-79
PGLOOP.FOR       9 12-Feb-79     WRTLP .FOR       2 12-Feb-79
SIEEE .OBJ       2 26-Apr-79     RIEEE .OBJ       3 26-Apr-79
SIEEE .SAV       3 26-Apr-79     RIEEE .SAV       3 26-Apr-79
PGLOOP.SAV      30               PGLOOP.OBJ      52
WRTLP .OBJ      12               PGMAC .OBJ       3
 14 Files, 162 Blocks
 318 Free Blocks
```

```
01-Jul-79
DXMNFB.SYS      97 18-Oct-78      PIP    .SAV      16 18-Oct-78
DX     .SYS      2 18-Oct-78      SYSLIB.OBJ     203 27-Dec-78
NL     .SYS      2 01-Feb-79      STARTF.COM       1
LINK   .SAV     29 18-Oct-78
 7 Files, 350 Blocks
 130 Free Blocks
```

```
01-Jul-79
NODAL   .OBJ    13 17-May-79    NODAL1.OBJ   57 17-May-79
FDM     .OBJ     5 17-May-79    M0000 .OBJ   46 17-May-79
M1000   .OBJ    17 17-May-79    M2000 .OBJ   16 17-May-79
M3000   .OBJ    14 17-May-79    M4000 .OBJ   55 17-May-79
M5000   .OBJ    56 17-May-79    M6000 .OBJ   21 17-May-79
LNK24   .COM     1 17-May-79    MAP   .MAP   20 17-May-79
DSK18   .DIR     1 01-Jul-79
 13 Files, 340 Blocks
 140 Free blocks
```

```
decwriter program:     FLP1.DIR       23-Jul-79     13:46:45     Page 001


23-Jul-79
DXMNSJ.SYS      86 14-Aug-77      TT    .SYS       2 14-Aug-77
DP    .SYS       2 14-Aug-77      DT    .SYS       2 14-Aug-77
DX    .SYS       2 14-Aug-77      RF    .SYS       2 14-Aug-77
RK    .SYS       2 14-Aug-77      DM    .SYS       4 14-Aug-77
DS    .SYS       2 14-Aug-77      LP    .SYS       2 14-Aug-77
NL    .SYS       2 14-Aug-77      STARTS.COM       1 11-Jul-77
PIP   .SAV      15 14-Aug-77      DUP   .SAV      17 14-Aug-77
DIR   .SAV      17 14-Aug-77      SYSMAC.SML      37 14-Aug-77
EDIT  .SAV      21 14-Aug-77      MACRO .SAV      45 14-Aug-77
CREF  .SAV       6 14-Aug-77      LINK  .SAV      29 16-Aug-77
LIBR  .SAV      18 14-Aug-77      FILEX .SAV      18 14-Aug-77
SRCCOM.SAV      11 14-Aug-77      DUMP  .SAV       7 14-Aug-77
PATCH .SAV       9 14-Aug-77      HELP  .SAV      21 14-Aug-77
HELP  .TEC       3 23-Jun-77      HELP  .TXT      73 19-Jul-77
DEMOBG.MAC       4 01-Aug-77      DEMOFG.MAC       5 01-Aug-77
V2USER.TXT       2 13-Jul-77      OTS1  .BAK       1
OTS1  .COM       1
 33 Files, 470 Blocks
 10 Free blocks
```

```
decrwiter program:     FLP2.DIR     23-Jul-79     13:47:31     Page 001


23-Jul-79
RKMNSJ.SYS      86 14-Aug-77      RKMNFB.SYS      96 14-Aug-77
RKMNXM.SYS     106 14-Aug-77      RKMNSJ.BL       82 14-Aug-77
DMMNSJ.SYS      87 14-Aug-77      ODT    .OBJ      9 14-Aug-77
 6 Files, 466 Blocks
14 Free blocks
```

```
23-Jul-79
DPMNXM.SYS   106 14-Aug-77      RFMNSJ.SYS    86 14-Aug-77
RFMNFB.SYS    96 14-Aug-77      RFMNXM.SYS   106 14-Aug-77
DTX    .SYS    2 14-Aug-77      DPX    .SYS    2 14-Aug-77
DXX    .SYS    3 14-Aug-77      FRX    .SYS    2 14-Aug-77
RKX    .SYS    2 14-Aug-77      DMX    .SYS    4 14-Aug-77
DSX    .SYS    2 14-Aug-77      LPX    .SYS    2 14-Aug-77
CR     .SYS    3 14-Aug-77      CRX    .SYS    3 14-Aug-77
MT     .SYS    8 14-Aug-77      MTX    .SYS    9 14-Aug-77
MTHD   .SYS    3 14-Aug-77      MTHDX  .SYS    4 14-Aug-77
BA     .SYS    7 14-Aug-77      BAX    .SYS    7 14-Aug-77
TT     .MAC    9 14-Aug-77      DEMOED.TXT     1 09-May-77
FLP3   .DIR    2 23-Jul-79
 23 Files, 469 Blocks
 11 Free blocks
```

```
decwriter program:      FLP4.DIR      23-Jul-79      13:48:25      Page 001


23-Jul-79
DMMNFB.SYS      98 14-Aug-77      DMMXXM.SYS     108 14-Aug-77
DXMNFB.SYS      97 14-Aug-77      DXMNXM.SYS     107 14-Aug-77
SYE    .SAV     50 14-Aug-77      NLX    .SYS      2 14-Aug-77
PC     .SYS      2 14-Aug-77      PCX    .SYS      2 14-Aug-77
DEMOF1.FOR       2 28-Jan-77
 9 Files, 468 Blocks
12 Free blocks
```

```
decwriter program:     FLP5.DIR     23-Jul-79     13:48:43     Page 001


23-Jul-79
DXMNSJ.BL       83 14-Aug-77     DTMNSJ.SYS      86 14-Aug-77
DTMNFB.SYS      96 14-Aug-77     DTMNSJ.BL       82 14-Aug-77
DSMNSJ.SYS      86 14-Aug-77     RF     .MAC      6 14-Aug-77
DM     .MAC     19 14-Aug-77     DEMOSP.MAC      11 01-Aug-77
 8 Files, 469 blocks
11 Free blocks
```

23-Jul-79

DSMNFB.SYS	96	14-Aug-77	DSMNXM.SYS	106	14-Aug-77
OPMNSJ.SYS	86	14-Aug-77	DPMNFB.SYS	97	14-Aug-77
DP .MAC	9	14-Aug-77	MUBRTE.OBJ	1	04-May-77
MUBTAB.OBJ	1	04-May-77	MUBZN1.OBJ	1	04-May-77
CT .MAC	32	14-Aug-77	TM .MAC	24	14-Aug-77
MMHD .SYS	4	14-Aug-77	MMHDX .SYS	4	14-Aug-77
PAT .SAV	7	14-Aug-77			

13 Files, 468 Blocks
12 Free blocks

```
decwriter program:     FLP7DIR     23-Jul-79     13:49:23     Page 001


23-Jul-79
KMON   .MAC    118 14-Aug-77     USR    .MAC     59 14-Aug-77
RMONSJ.MAC     56 14-Aug-77     RMONFB.MAC    138 14-Aug-77
EL     .MAC     18 14-Aug-77     LP     .MAC      7 14-Aug-77
RK     .MAC      7 14-Aug-77     DT     .MAC      7 14-Aug-77
DS     .MAC      7 14-Aug-77     PC     .MAC      6 14-Aug-77
ERRUTL.SAV      6 14-Aug-77     MACFST.SAV     45 14-Aug-77
 12 Files, 474 Blocks
 6 Free blocks
```

23-Jul-79

KMOVLY.MAC	153	14-Aug-77	BSTRAP.MAC	41	14-Aug-77
MTTEMT.MAC	18	16-Aug-77	MTTINT.MAC	34	14-Aug-77
TJ .MAC	28	14-Aug-77	NL .MAC	3	14-Aug-77
DX .MAC	11	14-Aug-77	CR .MAC	14	14-Aug-77
FSM .MAC	31	14-Aug-77	BATCH .MAC	98	14-Aug-77
MUBET1.OBJ	1	04-May-77	MUBXT1.OBJ	1	04-May-77
MUBZ1 .OBJ	1	04-May-77	MUBS1D.OBJ	1	04-May-77
MUBLNK.COM	1	04-May-77	MM .SYS	9	14-Aug-77
MMX .SYS	9	14-Aug-77	BA .MAC	19	14-Aug-77

18 Files, 473 Blocks
7 Free Blocks

```
decwriter program:     FLP9.DIR     23-Jul-79     13:50:13     Page 001


23-Jul-79
SYSMAC.MAC      36 01-Aug-77      BATCH .SAV      25 14-Aug-77
MACBK .SAV      52 14-Aug-77      PSE   .SAV      13 14-Aug-77
SYSGEN.SAV      32 14-Aug-77      SYSGEN.CND      83 08-Aug-77
SYSTBL.CND      26 08-Aug-77      VTMAC .MAC       7 01-Aug-77
VTHDLR.OBJ       8 14-Aug-77      SYSF4 .OBJ      39 14-Aug-77
MDUP  .SAV       9 14-Aug-77      MDUP  .MM       48 14-Aug-77
MDUP  .MT       48 14-Aug-77      MBOOT .BOT       1 14-Aug-77
MSBOOT.BOT       3 14-Aug-77      STARTF.COM       1 10-Jun-77
STARTX.COM       1 10-Jun-77      SJ    .MAC       1 01-Aug-77
FB    .MAC       1 01-Aug-77      XM    .MAC       1 01-Aug-77
SYSDEV.MAC       1 24-Sep-76      CT    .SYS       5 14-Aug-77
CTX   .SYS       6 14-Aug-77      TECO  .SAV      27 14-Aug-77
DEMOX1.MAC      5 01-Aug-77
 25 Files, 479 Blocks
 1 Free Blocks
```

APPENDIX 3 Startup Procedures

MSCDM STARTUP PROCEDURES

1. Power up all equipment (User Manual)
2. Insert diskette #1 into drive 0
3. Type DX (Return), system will now boot strap DEC F/B operating system which is used for loading MSCDM
4. Type in Date and Time

   DA DD-MON-YY
   TI HH:MM:SS

5. Start loader by typing .R FDMLDR
6. Press clear switch
7. Select MODE 1 (Normal Load), the loader will ask you to insert Diskette with File SIGGEN.LDA

** So insert Diskette #2 and press return the loader will load nodes SIG, 21, 22, 23, 25, 26, 27 from this diskette. It will ask you to insert diskette with file NODE28.SAV. So now insert Diskette #2A and press return. Loader finishes loading and starting nodes, also clears the CRT screen.

8. Ensure all LED's are on (Redo if not).
9. Turn PDU DC power supply OFF then ON
10. Insert Diskette #3 into Drive 0
11. Insert Diskette #3A into Drive 1
12. Type DX (return)

    System Bootstraps DEC S/J Operating system which is used for the applications program (Node 24)

13. Type the following commands:

    - Copy STATUS.MAS STATUS.DAT
    - RESET
    - R NODE 24

14. MSCDM is now running, the CRT will not be used from now on.

Note 1: After you have typed "ABORT" on MSCDM's LA36, you will notice a "STOP--" appear on CRT and the PDU will return to operating system.

** If you made a file updates (Mode 4) you should invoke a backup command file, so to save these changes. You do so by typing.@Backup.

Note 2: Should system crash, you should recover the data bases before restarting. To do so typing. @Crash

Note 3: Remember you are running S/J operating system, so for any reason if you change diskette in drive 0, you must reboot the new operating system...

```
COPY CDIR.OLD CDIR.DAT
COPY TDIR.OLD TDIR.DAT
COPY CXTD.OLD CXTD.DAT
COPY TNXD.OLD TNXD.DAT
```

```
COPY CDIR.DAT CDIR.OLD
COPY TDIR.DAT TDIR.OLD
COPY CKTD.DAT CKTD.OLD
COPY TNKD.DAT TNKD.OLD
```

USING THE PDU PROGRAM DEVELOPMENT SYSTEM FOR SOFTWARE GENERATION

Start-Up Procedures

1. Turn on main breaker at rear of PDU cabinet.
2. Turn on CRT switch on (right side of CRT).
3. Insert an operating diskette #1 into drive (0) (left).
4. Turn on all (three) switches on front of PDU.
5. A "$" should appear on CRT, type "DX" (return).
6. System should then "Boot strap displaying startup commands".
7. Type in date (.DA dd-mon-yy).
8. Type in time (.TI hh:mm:ss).
9. System is now ready to be used as a program develop system.
10. Diskette #1 contains the EDITOR, FORTRAN, MACRO system files this disk will be used to do all system functions except linking of FORTRAN object files.
11. Diskette #4 is used as a linking disk. It is inserted into Drive 0 and invoked by a user command file to link your FORTRAN object files.

APPENDIX 4 Interface Data

MSCDM REGISTER BITS

OPREG/INTERFACE

```
LST
BIT 0- PRGM I/O
BIT 1- RD
BIT 2- RS
BIT 3- WD
BIT 4- WC
BIT 5- SET INT. REQ
```

LIU/COMMAND REGISTER

```
LST
BIT 0- SEL ACRAM
BIT 1- SEL LDACTR
BIT 2-
BIT 3- SEL RDBUFADR
BIT 4- SEL MODSTAT
BIT 5- SEL INOROUTBUF
BIT 6- SEL IN/OUT
BIT 7- SEL 0/1
```

LIU/STATUS BYTE(0)

```
LST
BIT 0- IB0 FULL
BIT 1- IB1 FULL
BIT 2- IB0 OV-FL
BIT 3- IB1 OV-FL
BIT 4- PRSW
BIT 5- BKSW
BIT 6- WT DETECT
BIT 7- PRSW LATCH
```

LIU/STATUS BYTE(1)

```
LST
BIT 0- CRC 0 OK
BIT 1- CRC 1 OK
BIT 2- OB0 FULL
BIT 3- OB1 FULL
BIT 4- BDSLNSWPR
BIT 5- BDSLNSWBK
BIT 6- WRITE COMMAND
```

AD-A078 392 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66158 SBIE-AD-E100 314 NL

4 OF 4
ADA
078392

END
DATE
FILMED
1 -80
DDC

MSCDM REGISTER BITS CONT.

LIU/MODSTAT BYTE

LST
BIT 0- SET OB0 FULL
BIT 1- SET OB1 FULL
BIT 2- SET BDSLNSW PRIMARY
BIT 3- SET BDSLNSW BACKUP
BIT 4- SET WRITE COMMAND
BIT 5- CLR BDSLNSW PRIMARY
BIT 6- CLR BDSLNSW BACKUP

BLIUI CSR REGISTER

LST
BIT 0- LIU CLEAR=1,NORMAL=0
BIT 5- TIME-OUT OCCURED DURING DMA XFER
BIT 6- ENABLE DONE BIT INTERRUPT
BIT 7- STATUS OF CURRENT I/O OPERATION
BIT14- ENABLE LIU INTERRUPTS
BIT15- LIU REQUESTS AN INTERRUPT

INTERFACE/LIU COMMANDS

COMMAND	MST	LST	DEC	N/C
WCR: RS(0)	00010001	00000000	4352	RS
WCR: RS(1)	00010001	10000000	4488	RS
WCR: RDBUFADR IN / 0	00010001	00001000	4360	RD
WCR: RDBUFADR IN / 1	00010001	10001000	4488	RD
WCR: RDBUFADR OUT / 0	00010001	01001000	4424	RD
WCR: RDBUFADR OUT / 1	00010001	11001000	4552	RD
WCR: MODSTAT	00010001	00010000	4368	WD
WCR: LDACTR	00010001	00000010	4354	WD
WCR: ACRAM	00010001	00000001	4353	RD/WD
WCR: INBUF 0	00010001	00100000	4384	RD/WD
WCR: INBUF 1	00010001	10100000	4512	RD/WD
WCR: OUTBUF 0	00010001	01100000	4448	RD/WD
WCR: OUTBUF 1	00010001	11100000	4576	RD/WD
RS : PRGM	00000101	00000000	1280	
RD : PRGM	00000011	00000000	768	* CSR
RD : DMA	00100010	00000000	8704	+ CSR
WD : PRGM	00001001	00000000	2304	* CSR
WD : DMA	00101000	00000000	10240	+ CSR
WC : PRGM	00010001	00000000	4352	CMD

WCR: WRITE COMMAND REGISTER
RS : READ STATUS
RD : READ DATA (PROGRAM I/O OR DMA)
WD : WRITE DATA(PROGRAM I/O OR DMA)
WC : WRITE COMMAND REGISTER LIU

MSCDM ACRAM DATA

NREAD=(6) 0110
DREAD=(4) 0100
NULL =(7) 0111
WTOKN=(0) 0000

LOCATION	DR	NR	WT
NODE 20: SIG	"012	00	000
NODE 21: SSCI	"004	00	255
NODE 22: VSQC	"003	00	255
NODE 23: DSQC	"006	00	255
NODE 24: DBMS	"005	00	255
NODE 25: OCRI	"007	00	255
NODE 26: BWBSA	"010	00	255
NODE 27: FIAC	"011	00	255
NODE 28: SDCA	"002	00	255

NOTE: sig interlocks to loop though NODE 26(DLV11).

MSCDM INTERFACE BOARD DATA

LABEL	TYPE	ADDRESS	VECTOR	BAUD	TRANS/RECV	NODE	B/P POS.
(A)	DLV11-F	177560	60/62	9600	EIA/EIA	23	3 A-B
(B)	DLV11	177560	60/62	2400	PAS/PAS	28	3 A-B
(C)	DLV11	177560	60/62	9600	ACT/ACT	24	B1 C-D
(D)	DLV11-F	177510	310/312	9600	EIA/EIA	SIG	2 C-D
(E)	DLV11	177560	60/62	300	ACT/ACT	25	3 A-B
(F)	DLV11-F	175610	300/302	9600	EIA/EIA	SIG	2 A-B
(G)	DLV11-F	177410	320/322	9600	EIA/EIA	SIG	3 A-B
(H)	DLV11-F	177560	60/62	9600	EIA/EIA	26	3 A-B
(I)	DLV11-F	177560	60/62	9600	EIA/EIA	22	3 A-B
(J)	DLV11	177560	60/62	300	ACT/ACT	SIG	4 C-D
(K)	DLV11-F	177560	60/62	9600	EIA/EIA	21	3 A-B
(L)	IBV11-A	177560	420	N/A	IEEE-488	27	3 A-B

APPENDIX 5

GLOSSARY OF ACRONYMS

The interdisciplinary nature of the present study is emphasized by the large number of different acronyms, from diverse sources, that appear in the discussion. The following is a partial list of some of the relevant acronyms that have been identified. It also serves as a glossary.

ACAS	AUTOVON Centralized Alarm System
ACOC	Area Communications Operations Center
ADM	Adaptive Delta Modulation
ADO	Burroughs Advanced Development Organization
ASC	Automatic Swtiching Center (AUTODIN)
ASCII	American Standard Code for Information Interchange
ASCC	AUTODIN Station Control Console
ASSC	AUTODIN Station Supervisory Console
ASU	Alarm Scanner Unit
ATEC	Automated Tech Control
AVIE	AUTOVON Information and Evaluation Network
BARS	Buffered Automatic Reporting System
BBSA	Baseband Signal Analysis

BDLC	Burroughs Data Link Control
BLIUI	Bus Loop Interface Unit Interface
BWBSA	Combined functions of BBSA and WBSA
CCI	Command and Control Interpreter
CPU	Central Processor Unit
CRT	Cathode Ray Tube
DCA	Defense Communications Agency
DCAOC	Defense Communications Agency Operations Center
DCEC	Defense Communications Engineering Center
DCS	Defense Communications System
DBMS	Data Base Management Service
DDMS	Digital Distortion Monitoring Subsystem
DMA	Direct Memory Access
DSQC	Digital Service Quality Control
ESM	Exploratory System Control Model
ESMD	Exploratory System Control Model Development
FDM	Feasibility Development Model
FIAC	Fault Isolation and Analysis Coordination
IO	Input/Output
LA-36	DEC Hard Copy Terminal
LIU	Loop Interface Unit
MSCDM	Modular System Control Development Model
OCRI	Operator Control and Report Interface
PDU	Program Development Unit
PROM	Programmable Read Only Memory

RAM	Random Access Memory
SDCA	Switch Data Collection/Analysis
VSQC	Voice Service Quality Control
WBSA	Wideband Signal Analysis
WT	Write Token

Burroughs Corporation

Federal and Special Systems Group

Paoli, Pennsylvania 19301

Printed in U.S. America